

Python and C++ Containers

0.4.0

Generated by Doxygen 1.9.1

1 Python and C++ Containers	1
1.1 Using This Library	2
1.1.1 C++ To Python	2
1.1.2 Python to C++	2
1.2 Usage	3
1.2.1 Code Generation	3
1.2.2 C++ Build Configuration	3
1.2.2.1 Source Inclusion	3
1.2.3 Python Extension Example	3
1.2.3.1 Testing	4
2 Source Code for Python Cpp Homogenous Containers	5
2.1 <code><tt>cpp/</tt></code> Files of Note	5
2.2 <code><tt>cpy/</tt></code> Files and Directories of Note	5
2.3 <code><tt>ext/</tt></code> Files of Note	6
2.4 <code><tt>py/</tt></code> Files of Note	6
3 Namespace Index	7
3.1 Namespace List	7
4 Hierarchical Index	9
4.1 Class Hierarchy	9
5 Class Index	11
5.1 Class List	11
6 File Index	13
6.1 File List	13
7 Namespace Documentation	15
7.1 Python_Cpp_Containers Namespace Reference	15
7.1.1 Detailed Description	29
7.1.2 Enumeration Type Documentation	29
7.1.2.1 ErrorReturnValue	29
7.1.3 Function Documentation	29
7.1.3.1 <code>cpp_bool_to_py_bool()</code>	29
7.1.3.2 <code>cpp_complex_to_py_complex()</code>	30
7.1.3.3 <code>cpp_double_to_py_float()</code>	30
7.1.3.4 <code>cpp_long_to_py_long()</code>	30
7.1.3.5 <code>cpp_std_list_like_to_py_list()</code> [1/2]	31
7.1.3.6 <code>cpp_std_list_like_to_py_list()</code> [2/2]	31
7.1.3.7 <code>cpp_std_list_like_to_py_list< bool >()</code> [1/2]	32
7.1.3.8 <code>cpp_std_list_like_to_py_list< bool >()</code> [2/2]	32
7.1.3.9 <code>cpp_std_list_like_to_py_list< CppCustomObject >()</code>	32

7.1.3.10 <code>cpp_std_list_like_to_py_list< double >()</code> [1/2]	32
7.1.3.11 <code>cpp_std_list_like_to_py_list< double >()</code> [2/2]	33
7.1.3.12 <code>cpp_std_list_like_to_py_list< long >()</code> [1/2]	33
7.1.3.13 <code>cpp_std_list_like_to_py_list< long >()</code> [2/2]	34
7.1.3.14 <code>cpp_std_list_like_to_py_list< std::complex< double > >()</code> [1/2]	34
7.1.3.15 <code>cpp_std_list_like_to_py_list< std::complex< double > >()</code> [2/2]	34
7.1.3.16 <code>cpp_std_list_like_to_py_list< std::string >()</code> [1/2]	35
7.1.3.17 <code>cpp_std_list_like_to_py_list< std::string >()</code> [2/2]	35
7.1.3.18 <code>cpp_std_list_like_to_py_list< std::u16string >()</code> [1/2]	35
7.1.3.19 <code>cpp_std_list_like_to_py_list< std::u16string >()</code> [2/2]	36
7.1.3.20 <code>cpp_std_list_like_to_py_list< std::u32string >()</code> [1/2]	36
7.1.3.21 <code>cpp_std_list_like_to_py_list< std::u32string >()</code> [2/2]	37
7.1.3.22 <code>cpp_std_list_like_to_py_list< std::vector< char > >()</code> [1/2]	37
7.1.3.23 <code>cpp_std_list_like_to_py_list< std::vector< char > >()</code> [2/2]	37
7.1.3.24 <code>cpp_std_list_like_to_py_tuple()</code> [1/2]	38
7.1.3.25 <code>cpp_std_list_like_to_py_tuple()</code> [2/2]	38
7.1.3.26 <code>cpp_std_list_like_to_py_tuple< bool >()</code> [1/2]	39
7.1.3.27 <code>cpp_std_list_like_to_py_tuple< bool >()</code> [2/2]	39
7.1.3.28 <code>cpp_std_list_like_to_py_tuple< double >()</code> [1/2]	39
7.1.3.29 <code>cpp_std_list_like_to_py_tuple< double >()</code> [2/2]	40
7.1.3.30 <code>cpp_std_list_like_to_py_tuple< long >()</code> [1/2]	40
7.1.3.31 <code>cpp_std_list_like_to_py_tuple< long >()</code> [2/2]	40
7.1.3.32 <code>cpp_std_list_like_to_py_tuple< std::complex< double > >()</code> [1/2]	41
7.1.3.33 <code>cpp_std_list_like_to_py_tuple< std::complex< double > >()</code> [2/2]	41
7.1.3.34 <code>cpp_std_list_like_to_py_tuple< std::string >()</code> [1/2]	42
7.1.3.35 <code>cpp_std_list_like_to_py_tuple< std::string >()</code> [2/2]	42
7.1.3.36 <code>cpp_std_list_like_to_py_tuple< std::u16string >()</code> [1/2]	42
7.1.3.37 <code>cpp_std_list_like_to_py_tuple< std::u16string >()</code> [2/2]	43
7.1.3.38 <code>cpp_std_list_like_to_py_tuple< std::u32string >()</code> [1/2]	43
7.1.3.39 <code>cpp_std_list_like_to_py_tuple< std::u32string >()</code> [2/2]	43
7.1.3.40 <code>cpp_std_list_like_to_py_tuple< std::vector< char > >()</code> [1/2]	44
7.1.3.41 <code>cpp_std_list_like_to_py_tuple< std::vector< char > >()</code> [2/2]	44
7.1.3.42 <code>cpp_std_map_like_to_py_dict()</code> [1/3]	45
7.1.3.43 <code>cpp_std_map_like_to_py_dict()</code> [2/3]	45
7.1.3.44 <code>cpp_std_map_like_to_py_dict()</code> [3/3]	45
7.1.3.45 <code>cpp_std_map_like_to_py_dict< std::map, bool, bool >()</code>	45
7.1.3.46 <code>cpp_std_map_like_to_py_dict< std::map, bool, double >()</code>	46
7.1.3.47 <code>cpp_std_map_like_to_py_dict< std::map, bool, long >()</code>	46
7.1.3.48 <code>cpp_std_map_like_to_py_dict< std::map, bool, std::complex< double > >()</code>	47
7.1.3.49 <code>cpp_std_map_like_to_py_dict< std::map, bool, std::string >()</code>	47
7.1.3.50 <code>cpp_std_map_like_to_py_dict< std::map, bool, std::u16string >()</code>	47
7.1.3.51 <code>cpp_std_map_like_to_py_dict< std::map, bool, std::u32string >()</code>	48

7.1.3.52 <code>cpp_std_map_like_to_py_dict< std::map, bool, std::vector< char > >()</code>	48
7.1.3.53 <code>cpp_std_map_like_to_py_dict< std::map, double, bool >()</code>	49
7.1.3.54 <code>cpp_std_map_like_to_py_dict< std::map, double, double >()</code>	49
7.1.3.55 <code>cpp_std_map_like_to_py_dict< std::map, double, long >()</code>	49
7.1.3.56 <code>cpp_std_map_like_to_py_dict< std::map, double, std::complex< double > >()</code> . .	50
7.1.3.57 <code>cpp_std_map_like_to_py_dict< std::map, double, std::string >()</code>	50
7.1.3.58 <code>cpp_std_map_like_to_py_dict< std::map, double, std::u16string >()</code>	50
7.1.3.59 <code>cpp_std_map_like_to_py_dict< std::map, double, std::u32string >()</code>	51
7.1.3.60 <code>cpp_std_map_like_to_py_dict< std::map, double, std::vector< char > >()</code>	51
7.1.3.61 <code>cpp_std_map_like_to_py_dict< std::map, long, bool >()</code>	52
7.1.3.62 <code>cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject >()</code>	52
7.1.3.63 <code>cpp_std_map_like_to_py_dict< std::map, long, double >()</code>	52
7.1.3.64 <code>cpp_std_map_like_to_py_dict< std::map, long, long >()</code>	53
7.1.3.65 <code>cpp_std_map_like_to_py_dict< std::map, long, std::complex< double > >()</code> . . .	53
7.1.3.66 <code>cpp_std_map_like_to_py_dict< std::map, long, std::string >()</code>	53
7.1.3.67 <code>cpp_std_map_like_to_py_dict< std::map, long, std::u16string >()</code>	54
7.1.3.68 <code>cpp_std_map_like_to_py_dict< std::map, long, std::u32string >()</code>	54
7.1.3.69 <code>cpp_std_map_like_to_py_dict< std::map, long, std::vector< char > >()</code>	54
7.1.3.70 <code>cpp_std_map_like_to_py_dict< std::map, std::complex< double >, bool >()</code> . . .	55
7.1.3.71 <code>cpp_std_map_like_to_py_dict< std::map, std::complex< double >, double >()</code> . .	55
7.1.3.72 <code>cpp_std_map_like_to_py_dict< std::map, std::complex< double >, long >()</code> . . .	56
7.1.3.73 <code>cpp_std_map_like_to_py_dict< std::map, std::string, bool >()</code>	56
7.1.3.74 <code>cpp_std_map_like_to_py_dict< std::map, std::string, double >()</code>	56
7.1.3.75 <code>cpp_std_map_like_to_py_dict< std::map, std::string, long >()</code>	57
7.1.3.76 <code>cpp_std_map_like_to_py_dict< std::map, std::string, std::complex< double > >()</code>	57
7.1.3.77 <code>cpp_std_map_like_to_py_dict< std::map, std::string, std::string >()</code>	58
7.1.3.78 <code>cpp_std_map_like_to_py_dict< std::map, std::string, std::u16string >()</code>	58
7.1.3.79 <code>cpp_std_map_like_to_py_dict< std::map, std::string, std::u32string >()</code>	58
7.1.3.80 <code>cpp_std_map_like_to_py_dict< std::map, std::string, std::vector< char > >()</code> . . .	59
7.1.3.81 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, bool >()</code>	59
7.1.3.82 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, double >()</code>	60
7.1.3.83 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, long >()</code>	60
7.1.3.84 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double > >()</code>	60
7.1.3.85 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string >()</code>	61
7.1.3.86 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string >()</code>	61
7.1.3.87 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string >()</code>	62
7.1.3.88 <code>cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > >()</code> .	62
7.1.3.89 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, bool >()</code>	62
7.1.3.90 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, double >()</code>	63
7.1.3.91 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, long >()</code>	63
7.1.3.92 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, std::complex< double > >()</code>	64

7.1.3.93 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, std::string >()</code>	64
7.1.3.94 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u16string >()</code>	64
7.1.3.95 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u32string >()</code>	65
7.1.3.96 <code>cpp_std_map_like_to_py_dict< std::map, std::u32string, std::vector< char > >()</code>	65
7.1.3.97 <code>cpp_std_map_like_to_py_dict< std::map, std::vector< char >, bool >()</code>	66
7.1.3.98 <code>cpp_std_map_like_to_py_dict< std::map, std::vector< char >, double >()</code>	66
7.1.3.99 <code>cpp_std_map_like_to_py_dict< std::map, std::vector< char >, long >()</code>	66
7.1.3.100 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool >()</code>	67
7.1.3.101 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, double >()</code>	67
7.1.3.102 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, long >()</code>	68
7.1.3.103 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::complex< double > >()</code>	68
7.1.3.104 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::string >()</code>	68
7.1.3.105 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string >()</code>	69
7.1.3.106 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string >()</code>	69
7.1.3.107 <code>cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > >()</code>	70
7.1.3.108 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, bool >()</code>	70
7.1.3.109 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, double >()</code>	70
7.1.3.110 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, long >()</code>	71
7.1.3.111 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, std::complex< double > >()</code>	71
7.1.3.112 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, std::string >()</code>	72
7.1.3.113 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, std::u16string >()</code>	72
7.1.3.114 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, std::u32string >()</code>	72
7.1.3.115 <code>cpp_std_map_like_to_py_dict< std::unordered_map, double, std::vector< char > >()</code>	73
7.1.3.116 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, bool >()</code>	73
7.1.3.117 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, double >()</code>	74
7.1.3.118 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, long >()</code>	74
7.1.3.119 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, std::complex< double > >()</code>	74
7.1.3.120 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string >()</code>	75
7.1.3.121 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string >()</code>	75
7.1.3.122 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u32string >()</code>	76
7.1.3.123 <code>cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > >()</code>	76
7.1.3.124 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, bool >()</code>	76
7.1.3.125 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, double >()</code>	77
7.1.3.126 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, long >()</code>	77
7.1.3.127 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, bool >()</code>	78
7.1.3.128 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, double >()</code>	78
7.1.3.129 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, long >()</code>	78

7.1.3.130 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::complex< double > >()</code>	79
7.1.3.131 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::string >()</code>	79
7.1.3.132 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::u16string >()</code>	80
7.1.3.133 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::u32string >()</code>	80
7.1.3.134 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::vector< char > >()</code>	80
7.1.3.135 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, bool >()</code>	81
7.1.3.136 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, double >()</code>	81
7.1.3.137 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, long >()</code>	82
7.1.3.138 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::complex< double > >()</code>	82
7.1.3.139 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::string >()</code>	82
7.1.3.140 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::u16string >()</code>	83
7.1.3.141 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::u32string >()</code>	83
7.1.3.142 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::vector< char > >()</code>	84
7.1.3.143 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, bool >()</code>	84
7.1.3.144 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, double >()</code>	84
7.1.3.145 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, long >()</code>	85
7.1.3.146 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::complex< double > >()</code>	85
7.1.3.147 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::string >()</code>	86
7.1.3.148 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u16string >()</code>	86
7.1.3.149 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u32string >()</code>	86
7.1.3.150 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::vector< char > >()</code>	87
7.1.3.151 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, bool >()</code>	87
7.1.3.152 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, double >()</code>	88
7.1.3.153 <code>cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, long >()</code>	88
7.1.3.154 <code>cpp_std_unordered_set_to_py_frozenset()</code>	88
7.1.3.155 <code>cpp_std_unordered_set_to_py_frozenset< bool >()</code>	89
7.1.3.156 <code>cpp_std_unordered_set_to_py_frozenset< double >()</code>	89
7.1.3.157 <code>cpp_std_unordered_set_to_py_frozenset< long >()</code>	90
7.1.3.158 <code>cpp_std_unordered_set_to_py_frozenset< std::complex< double > >()</code>	90
7.1.3.159 <code>cpp_std_unordered_set_to_py_frozenset< std::string >()</code>	90
7.1.3.160 <code>cpp_std_unordered_set_to_py_frozenset< std::u16string >()</code>	91
7.1.3.161 <code>cpp_std_unordered_set_to_py_frozenset< std::u32string >()</code>	91
7.1.3.162 <code>cpp_std_unordered_set_to_py_frozenset< std::vector< char > >()</code>	92
7.1.3.163 <code>cpp_std_unordered_set_to_py_set()</code>	92

7.1.3.164 <code>cpp_std_unordered_set_to_py_set< bool >()</code>	92
7.1.3.165 <code>cpp_std_unordered_set_to_py_set< double >()</code>	93
7.1.3.166 <code>cpp_std_unordered_set_to_py_set< long >()</code>	93
7.1.3.167 <code>cpp_std_unordered_set_to_py_set< std::complex< double > >()</code>	94
7.1.3.168 <code>cpp_std_unordered_set_to_py_set< std::string >()</code>	94
7.1.3.169 <code>cpp_std_unordered_set_to_py_set< std::u16string >()</code>	94
7.1.3.170 <code>cpp_std_unordered_set_to_py_set< std::u32string >()</code>	95
7.1.3.171 <code>cpp_std_unordered_set_to_py_set< std::vector< char > >()</code>	95
7.1.3.172 <code>cpp_string_to_py_bytearray()</code>	95
7.1.3.173 <code>cpp_string_to_py_unicode8()</code>	96
7.1.3.174 <code>cpp_u16string_to_py_unicode16()</code>	96
7.1.3.175 <code>cpp_u32string_to_py_unicode32()</code>	97
7.1.3.176 <code>cpp_vector_char_to_py_bytearray()</code>	97
7.1.3.177 <code>cpp_vector_char_to_py_bytes()</code>	98
7.1.3.178 <code>generic_cpp_std_list_like_to_py_list()</code> [1/2]	98
7.1.3.179 <code>generic_cpp_std_list_like_to_py_list()</code> [2/2]	99
7.1.3.180 <code>generic_cpp_std_list_like_to_py_list_like()</code>	99
7.1.3.181 <code>generic_cpp_std_list_like_to_py_tuple()</code> [1/2]	99
7.1.3.182 <code>generic_cpp_std_list_like_to_py_tuple()</code> [2/2]	100
7.1.3.183 <code>generic_cpp_std_list_to_py_list()</code>	101
7.1.3.184 <code>generic_cpp_std_list_to_py_tuple()</code>	101
7.1.3.185 <code>generic_cpp_std_map_like_to_py_dict()</code>	101
7.1.3.186 <code>generic_cpp_std_unordered_set_to_py_frozenset()</code>	102
7.1.3.187 <code>generic_cpp_std_unordered_set_to_py_set()</code>	102
7.1.3.188 <code>generic_cpp_std_unordered_set_to_py_set_or_frozenset()</code>	103
7.1.3.189 <code>generic_cpp_std_vector_to_py_list()</code>	103
7.1.3.190 <code>generic_cpp_std_vector_to_py_tuple()</code>	103
7.1.3.191 <code>generic_py_dict_to_cpp_std_map_like()</code>	104
7.1.3.192 <code>generic_py_frozenset_to_cpp_std_unordered_set()</code>	104
7.1.3.193 <code>generic_py_list_to_cpp_std_list()</code>	105
7.1.3.194 <code>generic_py_list_to_cpp_std_list_like()</code> [1/2]	105
7.1.3.195 <code>generic_py_list_to_cpp_std_list_like()</code> [2/2]	106
7.1.3.196 <code>generic_py_list_to_cpp_std_vector()</code>	106
7.1.3.197 <code>generic_py_set_or_frozenset_to_cpp_std_unordered_set()</code>	106
7.1.3.198 <code>generic_py_set_to_cpp_std_unordered_set()</code>	107
7.1.3.199 <code>generic_py_tuple_to_cpp_std_list()</code>	107
7.1.3.200 <code>generic_py_tuple_to_cpp_std_list_like()</code> [1/2]	108
7.1.3.201 <code>generic_py_tuple_to_cpp_std_list_like()</code> [2/2]	108
7.1.3.202 <code>generic_py_tuple_to_cpp_std_vector()</code>	109
7.1.3.203 <code>generic_py_unary_to_cpp_std_list_like()</code>	109
7.1.3.204 <code>py_bool_check()</code>	109
7.1.3.205 <code>py_bool_to_cpp_bool()</code>	109

7.1.3.206 <code>py_bytearray_check()</code>	110
7.1.3.207 <code>py_bytearray_to_cpp_string()</code>	110
7.1.3.208 <code>py_bytearray_to_cpp_vector_char()</code>	110
7.1.3.209 <code>py_bytes_check()</code>	111
7.1.3.210 <code>py_bytes_to_cpp_vector_char()</code>	111
7.1.3.211 <code>py_complex_check()</code>	111
7.1.3.212 <code>py_complex_to_cpp_complex()</code>	112
7.1.3.213 <code>py_dict_to_cpp_std_map_like()</code> [1/3]	112
7.1.3.214 <code>py_dict_to_cpp_std_map_like()</code> [2/3]	113
7.1.3.215 <code>py_dict_to_cpp_std_map_like()</code> [3/3]	113
7.1.3.216 <code>py_dict_to_cpp_std_map_like< std::map, bool, bool >()</code>	113
7.1.3.217 <code>py_dict_to_cpp_std_map_like< std::map, bool, double >()</code>	113
7.1.3.218 <code>py_dict_to_cpp_std_map_like< std::map, bool, long >()</code>	114
7.1.3.219 <code>py_dict_to_cpp_std_map_like< std::map, bool, std::complex< double > >()</code>	114
7.1.3.220 <code>py_dict_to_cpp_std_map_like< std::map, bool, std::string >()</code>	115
7.1.3.221 <code>py_dict_to_cpp_std_map_like< std::map, bool, std::u16string >()</code>	115
7.1.3.222 <code>py_dict_to_cpp_std_map_like< std::map, bool, std::u32string >()</code>	116
7.1.3.223 <code>py_dict_to_cpp_std_map_like< std::map, bool, std::vector< char > >()</code>	116
7.1.3.224 <code>py_dict_to_cpp_std_map_like< std::map, double, bool >()</code>	117
7.1.3.225 <code>py_dict_to_cpp_std_map_like< std::map, double, double >()</code>	117
7.1.3.226 <code>py_dict_to_cpp_std_map_like< std::map, double, long >()</code>	117
7.1.3.227 <code>py_dict_to_cpp_std_map_like< std::map, double, std::complex< double > >()</code>	118
7.1.3.228 <code>py_dict_to_cpp_std_map_like< std::map, double, std::string >()</code>	118
7.1.3.229 <code>py_dict_to_cpp_std_map_like< std::map, double, std::u16string >()</code>	119
7.1.3.230 <code>py_dict_to_cpp_std_map_like< std::map, double, std::u32string >()</code>	119
7.1.3.231 <code>py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > >()</code>	120
7.1.3.232 <code>py_dict_to_cpp_std_map_like< std::map, long, bool >()</code>	120
7.1.3.233 <code>py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject >()</code>	120
7.1.3.234 <code>py_dict_to_cpp_std_map_like< std::map, long, double >()</code>	121
7.1.3.235 <code>py_dict_to_cpp_std_map_like< std::map, long, long >()</code>	121
7.1.3.236 <code>py_dict_to_cpp_std_map_like< std::map, long, std::complex< double > >()</code>	121
7.1.3.237 <code>py_dict_to_cpp_std_map_like< std::map, long, std::string >()</code>	122
7.1.3.238 <code>py_dict_to_cpp_std_map_like< std::map, long, std::u16string >()</code>	122
7.1.3.239 <code>py_dict_to_cpp_std_map_like< std::map, long, std::u32string >()</code>	123
7.1.3.240 <code>py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > >()</code>	123
7.1.3.241 <code>py_dict_to_cpp_std_map_like< std::map, std::complex< double >, bool >()</code>	124
7.1.3.242 <code>py_dict_to_cpp_std_map_like< std::map, std::complex< double >, double >()</code>	124
7.1.3.243 <code>py_dict_to_cpp_std_map_like< std::map, std::complex< double >, long >()</code>	124
7.1.3.244 <code>py_dict_to_cpp_std_map_like< std::map, std::string, bool >()</code>	125
7.1.3.245 <code>py_dict_to_cpp_std_map_like< std::map, std::string, double >()</code>	125
7.1.3.246 <code>py_dict_to_cpp_std_map_like< std::map, std::string, long >()</code>	126
7.1.3.247 <code>py_dict_to_cpp_std_map_like< std::map, std::string, std::complex< double > >()</code>	126

7.1.3.248 py_dict_to_cpp_std_map_like< std::map, std::string, std::string >()	127
7.1.3.249 py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string >()	127
7.1.3.250 py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string >()	128
7.1.3.251 py_dict_to_cpp_std_map_like< std::map, std::string, std::vector< char > >()	128
7.1.3.252 py_dict_to_cpp_std_map_like< std::map, std::u16string, bool >()	128
7.1.3.253 py_dict_to_cpp_std_map_like< std::map, std::u16string, double >()	130
7.1.3.254 py_dict_to_cpp_std_map_like< std::map, std::u16string, long >()	130
7.1.3.255 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex< double > >()	131
7.1.3.256 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string >()	131
7.1.3.257 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string >()	132
7.1.3.258 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string >()	132
7.1.3.259 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector< char > >()	132
7.1.3.260 py_dict_to_cpp_std_map_like< std::map, std::u32string, bool >()	133
7.1.3.261 py_dict_to_cpp_std_map_like< std::map, std::u32string, double >()	133
7.1.3.262 py_dict_to_cpp_std_map_like< std::map, std::u32string, long >()	134
7.1.3.263 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex< double > >()	134
7.1.3.264 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string >()	135
7.1.3.265 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string >()	135
7.1.3.266 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string >()	136
7.1.3.267 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector< char > >()	136
7.1.3.268 py_dict_to_cpp_std_map_like< std::map, std::vector< char >, bool >()	136
7.1.3.269 py_dict_to_cpp_std_map_like< std::map, std::vector< char >, double >()	137
7.1.3.270 py_dict_to_cpp_std_map_like< std::map, std::vector< char >, long >()	137
7.1.3.271 py_dict_to_cpp_std_map_like< std::unordered_map, bool, bool >()	138
7.1.3.272 py_dict_to_cpp_std_map_like< std::unordered_map, bool, double >()	138
7.1.3.273 py_dict_to_cpp_std_map_like< std::unordered_map, bool, long >()	139
7.1.3.274 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::complex< double > >()	139
7.1.3.275 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::string >()	140
7.1.3.276 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u16string >()	140
7.1.3.277 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u32string >()	140
7.1.3.278 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::vector< char > >()	141
7.1.3.279 py_dict_to_cpp_std_map_like< std::unordered_map, double, bool >()	141
7.1.3.280 py_dict_to_cpp_std_map_like< std::unordered_map, double, double >()	142
7.1.3.281 py_dict_to_cpp_std_map_like< std::unordered_map, double, long >()	142
7.1.3.282 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::complex< dou- ble > >()	143
7.1.3.283 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::string >()	143
7.1.3.284 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u16string >()	144
7.1.3.285 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u32string >()	144

7.1.3.286 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::vector< char > >()	144
7.1.3.287 py_dict_to_cpp_std_map_like< std::unordered_map, long, bool >()	145
7.1.3.288 py_dict_to_cpp_std_map_like< std::unordered_map, long, double >()	145
7.1.3.289 py_dict_to_cpp_std_map_like< std::unordered_map, long, long >()	146
7.1.3.290 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex< double > >()	146
7.1.3.291 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::string >()	147
7.1.3.292 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u16string >()	147
7.1.3.293 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u32string >()	148
7.1.3.294 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::vector< char > >()	148
7.1.3.295 py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, bool >()	148
7.1.3.296 py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, double >()	149
7.1.3.297 py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, long >()	149
7.1.3.298 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, bool >()	150
7.1.3.299 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, double >()	150
7.1.3.300 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, long >()	151
7.1.3.301 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::complex< double > >()	151
7.1.3.302 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::string >()	152
7.1.3.303 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u16string >()	152
7.1.3.304 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u32string >()	152
7.1.3.305 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::vector< char > >()	153
7.1.3.306 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, bool >()	153
7.1.3.307 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, double >()	154
7.1.3.308 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, long >()	154
7.1.3.309 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::complex< double > >()	155
7.1.3.310 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::string >()	155
7.1.3.311 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u16string >()	156
7.1.3.312 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u32string >()	156
7.1.3.313 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::vector< char > >()	156
7.1.3.314 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, bool >()	157
7.1.3.315 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, double >()	157
7.1.3.316 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, long >()	158
7.1.3.317 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::complex< double > >()	158
7.1.3.318 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::string >()	159

7.1.3.319 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u16string >()	159
7.1.3.320 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u32string >()	160
7.1.3.321 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::vector< char > >()	160
7.1.3.322 py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, bool >()	160
7.1.3.323 py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, double >()	161
7.1.3.324 py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, long >()	161
7.1.3.325 py_float_check()	162
7.1.3.326 py_float_to_cpp_double()	162
7.1.3.327 py_frozenset_check()	163
7.1.3.328 py_frozenset_to_cpp_std_unordered_set()	163
7.1.3.329 py_frozenset_to_cpp_std_unordered_set< bool >()	163
7.1.3.330 py_frozenset_to_cpp_std_unordered_set< double >()	164
7.1.3.331 py_frozenset_to_cpp_std_unordered_set< long >()	164
7.1.3.332 py_frozenset_to_cpp_std_unordered_set< std::complex< double > >()	165
7.1.3.333 py_frozenset_to_cpp_std_unordered_set< std::string >()	165
7.1.3.334 py_frozenset_to_cpp_std_unordered_set< std::u16string >()	165
7.1.3.335 py_frozenset_to_cpp_std_unordered_set< std::u32string >()	166
7.1.3.336 py_frozenset_to_cpp_std_unordered_set< std::vector< char > >()	166
7.1.3.337 py_list_check()	167
7.1.3.338 py_list_get()	167
7.1.3.339 py_list_len()	168
7.1.3.340 py_list_new()	168
7.1.3.341 py_list_set()	168
7.1.3.342 py_list_to_cpp_std_list_like() [1/2]	169
7.1.3.343 py_list_to_cpp_std_list_like() [2/2]	169
7.1.3.344 py_list_to_cpp_std_list_like< bool >() [1/2]	170
7.1.3.345 py_list_to_cpp_std_list_like< bool >() [2/2]	170
7.1.3.346 py_list_to_cpp_std_list_like< CppCustomObject >()	170
7.1.3.347 py_list_to_cpp_std_list_like< double >() [1/2]	171
7.1.3.348 py_list_to_cpp_std_list_like< double >() [2/2]	171
7.1.3.349 py_list_to_cpp_std_list_like< long >() [1/2]	171
7.1.3.350 py_list_to_cpp_std_list_like< long >() [2/2]	172
7.1.3.351 py_list_to_cpp_std_list_like< std::complex< double > >() [1/2]	172
7.1.3.352 py_list_to_cpp_std_list_like< std::complex< double > >() [2/2]	173
7.1.3.353 py_list_to_cpp_std_list_like< std::string >() [1/2]	173
7.1.3.354 py_list_to_cpp_std_list_like< std::string >() [2/2]	173
7.1.3.355 py_list_to_cpp_std_list_like< std::u16string >() [1/2]	174
7.1.3.356 py_list_to_cpp_std_list_like< std::u16string >() [2/2]	174
7.1.3.357 py_list_to_cpp_std_list_like< std::u32string >() [1/2]	175

7.1.3.358 py_list_to_cpp_std_list_like< std::u32string >() [2/2]	175
7.1.3.359 py_list_to_cpp_std_list_like< std::vector< char > >() [1/2]	175
7.1.3.360 py_list_to_cpp_std_list_like< std::vector< char > >() [2/2]	176
7.1.3.361 py_long_check()	176
7.1.3.362 py_long_to_cpp_long()	177
7.1.3.363 py_set_check()	177
7.1.3.364 py_set_to_cpp_std_unordered_set()	177
7.1.3.365 py_set_to_cpp_std_unordered_set< bool >()	178
7.1.3.366 py_set_to_cpp_std_unordered_set< double >()	178
7.1.3.367 py_set_to_cpp_std_unordered_set< long >()	179
7.1.3.368 py_set_to_cpp_std_unordered_set< std::complex< double > >()	179
7.1.3.369 py_set_to_cpp_std_unordered_set< std::string >()	180
7.1.3.370 py_set_to_cpp_std_unordered_set< std::u16string >()	180
7.1.3.371 py_set_to_cpp_std_unordered_set< std::u32string >()	180
7.1.3.372 py_set_to_cpp_std_unordered_set< std::vector< char > >()	181
7.1.3.373 py_tuple_check()	181
7.1.3.374 py_tuple_get()	182
7.1.3.375 py_tuple_len()	182
7.1.3.376 py_tuple_new()	182
7.1.3.377 py_tuple_set()	183
7.1.3.378 py_tuple_to_cpp_std_list_like() [1/2]	183
7.1.3.379 py_tuple_to_cpp_std_list_like() [2/2]	184
7.1.3.380 py_tuple_to_cpp_std_list_like< bool >() [1/2]	184
7.1.3.381 py_tuple_to_cpp_std_list_like< bool >() [2/2]	185
7.1.3.382 py_tuple_to_cpp_std_list_like< double >() [1/2]	185
7.1.3.383 py_tuple_to_cpp_std_list_like< double >() [2/2]	186
7.1.3.384 py_tuple_to_cpp_std_list_like< long >() [1/2]	186
7.1.3.385 py_tuple_to_cpp_std_list_like< long >() [2/2]	186
7.1.3.386 py_tuple_to_cpp_std_list_like< std::complex< double > >() [1/2]	187
7.1.3.387 py_tuple_to_cpp_std_list_like< std::complex< double > >() [2/2]	187
7.1.3.388 py_tuple_to_cpp_std_list_like< std::string >() [1/2]	188
7.1.3.389 py_tuple_to_cpp_std_list_like< std::string >() [2/2]	188
7.1.3.390 py_tuple_to_cpp_std_list_like< std::u16string >() [1/2]	188
7.1.3.391 py_tuple_to_cpp_std_list_like< std::u16string >() [2/2]	189
7.1.3.392 py_tuple_to_cpp_std_list_like< std::u32string >() [1/2]	189
7.1.3.393 py_tuple_to_cpp_std_list_like< std::u32string >() [2/2]	190
7.1.3.394 py_tuple_to_cpp_std_list_like< std::vector< char > >() [1/2]	190
7.1.3.395 py_tuple_to_cpp_std_list_like< std::vector< char > >() [2/2]	190
7.1.3.396 py_unicode16_check()	191
7.1.3.397 py_unicode16_to_cpp_u16string()	191
7.1.3.398 py_unicode32_check()	192
7.1.3.399 py_unicode32_to_cpp_u32string()	192

7.1.3.400 py_unicode8_check()	192
7.1.3.401 py_unicode8_to_cpp_string()	193
7.1.3.402 very_generic_cpp_std_list_like_to_py_unary()	193
7.1.3.403 very_generic_py_unary_to_cpp_std_list_like()	194
7.2 src Namespace Reference	195
7.3 src.py Namespace Reference	195
7.4 src.py.code_gen Namespace Reference	195
7.4.1 Detailed Description	196
7.4.2 Function Documentation	196
7.4.2.1 declarations()	197
7.4.2.2 definitions()	197
7.4.2.3 defn_name_from_decl_name()	197
7.4.2.4 dict_map_declarations()	197
7.4.2.5 dict_map_definitions()	197
7.4.2.6 main()	198
7.4.2.7 unary_declarations()	198
7.4.2.8 unary_definitions()	198
7.4.2.9 write_files()	198
7.4.3 Variable Documentation	198
7.4.3.1 AUTO_FILE_NAME	198
7.4.3.2 CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL	198
7.4.3.3 CPP_MAP_TYPE_TO_PY_DICT_DECL	199
7.4.3.4 CPP_MAP_TYPE_TO_PY_DICT_DEFN	199
7.4.3.5 CPP_MAP_TYPES	199
7.4.3.6 CPP_NAMESPACE	199
7.4.3.7 CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL	199
7.4.3.8 CPP_PY_DICT_TO_MAP_TYPE_DECL	200
7.4.3.9 CPP_PY_DICT_TO_MAP_TYPE_DEFN	200
7.4.3.10 CPP_TYPE_TO_FUNCS	200
7.4.3.11 CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINER	201
7.4.3.12 CPP_UNARY_FUNCTION_TO_PY_BASE_DECL	201
7.4.3.13 CPP_UNARY_FUNCTION_TO_PY_DECL	201
7.4.3.14 CPP_UNARY_FUNCTION_TO_PY_DEFN	201
7.4.3.15 logger	201
7.4.3.16 PROJECT_VERSION	202
7.4.3.17 PY_TO_CPP_UNARY_FUNCTION_BASE_DECL	202
7.4.3.18 PY_TO_CPP_UNARY_FUNCTION_DECL	202
7.4.3.19 PY_TO_CPP_UNARY_FUNCTION_DEFN	202
7.4.3.20 REQUIRED_INCLUDES	202
7.4.3.21 UNARY_COLLECTIONS	203
7.5 src.py.code_gen_common Namespace Reference	203
7.6 src.py.code_gen_documentation Namespace Reference	203

7.6.1 Detailed Description	204
7.6.2 Function Documentation	204
7.6.2.1 comment_list_str()	204
7.6.2.2 comment_str()	204
7.6.2.3 cpp_comment_section()	204
7.6.2.4 documentation()	204
7.6.2.5 doxygen_cpp_to_python_dict_base_class()	205
7.6.2.6 doxygen_cpp_to_python_dict_instantiation()	205
7.6.2.7 doxygen_cpp_to_python_unary_base_class()	205
7.6.2.8 doxygen_cpp_to_python_unary_instantiation()	206
7.6.2.9 doxygen_python_dict_to_cpp_base_class()	206
7.6.2.10 doxygen_python_dict_to_cpp_instantiation()	206
7.6.2.11 doxygen_python_to_cpp_unary_base_class()	207
7.6.2.12 doxygen_python_to_cpp_unary_instantiation()	207
7.6.2.13 get_codegen_please_no_edit_warning()	208
7.6.2.14 get_codegen_please_no_edit_warning_context()	208
7.6.3 Variable Documentation	208
7.6.3.1 WIDTH	208
7.7 std Namespace Reference	208
8 Class Documentation	209
8.1 src.py.code_gen.CodeCount Class Reference	209
8.1.1 Detailed Description	209
8.2 CppCustomObject Class Reference	209
8.2.1 Constructor & Destructor Documentation	210
8.2.1.1 CppCustomObject() [1/2]	210
8.2.1.2 CppCustomObject() [2/2]	210
8.2.2 Member Function Documentation	210
8.2.2.1 first()	210
8.2.2.2 last()	210
8.2.2.3 name()	210
8.2.2.4 number()	210
8.3 src.py.code_gen_common.CppTypeFunctions Class Reference	211
8.3.1 Detailed Description	211
8.4 CustomObject Struct Reference	211
8.4.1 Member Data Documentation	211
8.4.1.1 first	211
8.4.1.2 last	211
8.4.1.3 number	212
8.5 ExecClock Class Reference	212
8.5.1 Member Typedef Documentation	212
8.5.1.1 tHiResDouble	212

8.5.2 Constructor & Destructor Documentation	212
8.5.2.1 ExecClock()	212
8.5.3 Member Function Documentation	212
8.5.3.1 seconds()	213
8.6 std::hash< std::complex< double > > Struct Reference	213
8.6.1 Detailed Description	213
8.6.2 Member Function Documentation	213
8.6.2.1 operator()()	213
8.7 std::hash< std::vector< char > > Struct Reference	213
8.7.1 Detailed Description	214
8.7.2 Member Function Documentation	214
8.7.2.1 operator()()	214
8.8 std::less< std::complex< T > > Struct Template Reference	214
8.8.1 Member Function Documentation	214
8.8.1.1 operator()()	214
8.9 RSSSnapshot Class Reference	214
8.9.1 Detailed Description	215
8.9.2 Constructor & Destructor Documentation	215
8.9.2.1 RSSSnapshot()	215
8.9.3 Member Function Documentation	215
8.9.3.1 name()	215
8.9.3.2 rss_initial_mb()	216
8.9.3.3 rss_initial_pages()	216
8.9.3.4 rss_now_diff_mb()	216
8.9.3.5 rss_now_diff_pages()	216
8.9.3.6 rss_now_mb()	216
8.9.3.7 rss_now_pages()	216
8.9.3.8 rss_peak_diff_mb()	216
8.9.3.9 rss_peak_diff_pages()	216
8.9.3.10 rss_peak_initial_mb()	217
8.9.3.11 rss_peak_initial_pages()	217
8.9.3.12 rss_peak_now_mb()	217
8.9.3.13 rss_peak_now_pages()	217
8.9.4 Member Data Documentation	217
8.9.4.1 m_name	217
8.9.4.2 m_rss_initial	217
8.9.4.3 m_rss_peak_initial	217
8.10 StreamFormatState Class Reference	218
8.10.1 Detailed Description	218
8.10.2 Constructor & Destructor Documentation	218
8.10.2.1 StreamFormatState()	218
8.10.2.2 ~StreamFormatState()	218

8.11 SubTestCount Class Reference	219
8.11.1 Detailed Description	219
8.11.2 Constructor & Destructor Documentation	219
8.11.2.1 SubTestCount()	219
8.11.3 Member Function Documentation	219
8.11.3.1 failure()	219
8.11.3.2 test()	220
8.11.3.3 test_count()	220
8.11.3.4 test_failures()	220
8.11.4 Member Data Documentation	220
8.11.4.1 m_failure	220
8.11.4.2 m_test_count	220
8.12 TestResult Class Reference	220
8.12.1 Detailed Description	221
8.12.1.1 Note on gnuplot	222
8.12.1.2 Note on Using the Rate Column	223
8.12.2 Constructor & Destructor Documentation	223
8.12.2.1 TestResult() [1/4]	223
8.12.2.2 TestResult() [2/4]	224
8.12.2.3 TestResult() [3/4]	224
8.12.2.4 TestResult() [4/4]	224
8.12.3 Member Function Documentation	224
8.12.3.1 atomicTestMeanExecTime()	224
8.12.3.2 execTime()	224
8.12.3.3 execTimeAdd()	224
8.12.3.4 execTimeMax()	225
8.12.3.5 execTimeMin()	225
8.12.3.6 execTimeStdDev()	225
8.12.3.7 failed() [1/2]	225
8.12.3.8 failed() [2/2]	225
8.12.3.9 hasExecTimeStdDev()	225
8.12.3.10 name()	225
8.12.3.11 numScaleValues()	226
8.12.3.12 numTests() [1/2]	226
8.12.3.13 numTests() [2/2]	226
8.12.3.14 operator=()	226
8.12.3.15 scaleValues()	226
8.12.3.16 setFailed()	226
8.12.3.17 testCount()	226
8.12.3.18 totalTime()	227
8.13 TestResultS Class Reference	227
8.13.1 Detailed Description	227

8.13.2 Member Typedef Documentation	227
8.13.2.1 tResults	227
8.13.3 Constructor & Destructor Documentation	227
8.13.3.1 TestResultS()	228
8.13.4 Member Function Documentation	228
8.13.4.1 dump_header()	228
8.13.4.2 dump_tail()	228
8.13.4.3 dump_tests()	228
8.13.4.4 failed()	228
8.13.4.5 push_back()	228
8.13.4.6 results()	229
8.14 src.py.code_gen_common.TypeConversionFunctions Class Reference	229
8.14.1 Detailed Description	229
8.15 src.py.code_gen_common.UnaryFunctions Class Reference	229
8.15.1 Detailed Description	229
9 File Documentation	231
9.1 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/__init__.py File Reference	231
9.2 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/__init__.py File Reference	231
9.3 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/get_rss.cpp File Reference	231
9.3.1 Macro Definition Documentation	232
9.3.1.1 RSS_SNAPSHOT_REPORT_PAGES	232
9.3.2 Function Documentation	232
9.3.2.1 getCurrentRSS()	232
9.3.2.2 getCurrentRSS_alterate()	232
9.3.2.3 getCurrentRSS_alterateMb()	233
9.3.2.4 getCurrentRSSMb()	233
9.3.2.5 getPeakRSS()	233
9.3.2.6 getPeakRSSMb()	233
9.3.2.7 operator<<()	233
9.3.3 Variable Documentation	233
9.3.3.1 MB_PRECISION	233
9.3.3.2 MB_WIDTH	233
9.3.3.3 MEGABYTES	234
9.4 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/get_rss.h File Reference	234
9.4.1 Function Documentation	234
9.4.1.1 getCurrentRSS()	234
9.4.1.2 getCurrentRSS_alterate()	235
9.4.1.3 getPeakRSS()	235
9.4.1.4 operator<<()	235
9.4.2 Variable Documentation	235
9.4.2.1 MEGABYTES	235

9.5	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/save_stream_state.h File Reference	235
9.6	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.cpp File Reference	236
9.6.1	Macro Definition Documentation	236
9.6.1.1	REGEX_SPACE_ANYTHING	237
9.6.1.2	REGEX_SPACE_FLOAT	237
9.6.1.3	REGEX_SPACE_INTEGER	237
9.6.1.4	REGEX_SPACE_STRING_NO_SPACE	237
9.6.2	Function Documentation	237
9.6.2.1	count_of_unique_string()	237
9.6.2.2	operator<<() [1/2]	237
9.6.2.3	Note on The Output	238
9.6.2.4	operator<<() [2/2]	239
9.6.2.5	reset_count_of_unique_string()	239
9.6.2.6	unique_string()	239
9.6.2.7	unique_u16string()	239
9.6.2.8	unique_u32string()	240
9.6.2.9	unique_vector_char()	240
9.6.3	Variable Documentation	240
9.6.3.1	str_count	241
9.6.3.2	TIME_PRECISION	241
9.6.3.3	TIME_WIDTH	241
9.7	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.h File Reference	241
9.7.1	Function Documentation	242
9.7.1.1	count_of_unique_string()	242
9.7.1.2	operator<<() [1/2]	242
9.7.1.3	Note on The Output	242
9.7.1.4	operator<<() [2/2]	243
9.7.1.5	reset_count_of_unique_string()	243
9.7.1.6	unique_string()	243
9.7.1.7	unique_u16string()	243
9.7.1.8	unique_u32string()	244
9.7.1.9	unique_vector_char()	244
9.8	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/auto_py_convert_↔ internal.cpp File Reference	244
9.9	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/auto_py_convert_↔ internal.h File Reference	256
9.10	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_container_↔ convert.cpp File Reference	268
9.11	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_container_↔ convert.h File Reference	268

9.12	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert.h File Reference	269
9.12.1	Macro Definition Documentation	271
9.12.1.1	PYTHON_CPP_CONTAINERS_VERSION	271
9.13	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert_↔scrap.h File Reference	271
9.14	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_↔convert.cpp File Reference	273
9.15	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_↔convert.h File Reference	274
9.16	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.cpp File Reference	275
9.16.1	Function Documentation	277
9.16.1.1	compare_dict< std::map, std::string, std::string >()	277
9.16.1.2	compare_dict< std::map, std::u16string, std::u16string >()	278
9.16.1.3	compare_dict< std::map, std::u32string, std::u32string >()	278
9.16.1.4	compare_dict< std::unordered_map, std::string, std::string >()	278
9.16.1.5	compare_dict< std::unordered_map, std::u16string, std::u16string >()	278
9.16.1.6	compare_dict< std::unordered_map, std::u32string, std::u32string >()	278
9.16.1.7	compare_list< bool >()	278
9.16.1.8	compare_list< double >()	279
9.16.1.9	compare_list< long >()	279
9.16.1.10	compare_list< std::complex< double > >()	279
9.16.1.11	compare_list< std::string >()	279
9.16.1.12	compare_list< std::u16string >()	279
9.16.1.13	compare_list< std::u32string >()	279
9.16.1.14	compare_list< std::vector< char > >()	280
9.16.1.15	compare_set< std::string >()	280
9.16.1.16	compare_set< std::u16string >()	280
9.16.1.17	compare_set< std::u32string >()	280
9.16.1.18	compare_set< std::vector< char > >()	280
9.16.1.19	compare_tuple< bool >()	280
9.16.1.20	compare_tuple< double >()	281
9.16.1.21	compare_tuple< long >()	281
9.16.1.22	compare_tuple< std::complex< double > >()	281
9.16.1.23	compare_tuple< std::string >()	281
9.16.1.24	compare_tuple< std::u16string >()	281
9.16.1.25	compare_tuple< std::u32string >()	281
9.16.1.26	compare_tuple< std::vector< char > >()	282
9.16.1.27	new_py_dict_bytes()	282
9.16.1.28	new_py_dict_string()	282
9.16.1.29	new_py_dict_string16()	283
9.16.1.30	new_py_dict_string32()	283

9.16.1.31 new_py_list_bytes()	283
9.16.1.32 new_py_list_string()	284
9.16.1.33 new_py_list_string16()	284
9.16.1.34 new_py_list_string32()	284
9.16.1.35 new_py_set_bytes()	285
9.16.1.36 new_py_set_string()	285
9.16.1.37 new_py_set_u16string()	286
9.16.1.38 new_py_set_u32string()	286
9.16.1.39 new_py_tuple_bytes()	286
9.16.1.40 new_py_tuple_string()	287
9.16.1.41 new_py_tuple_string16()	287
9.16.1.42 new_py_tuple_string32()	287
9.16.1.43 test_cpp_std_map_like_to_py_dict_bytes()	288
9.16.1.44 test_cpp_std_map_like_to_py_dict_string()	288
9.16.1.45 test_cpp_std_map_like_to_py_dict_string16()	288
9.16.1.46 test_cpp_std_map_like_to_py_dict_string32()	288
9.16.1.47 test_cpp_std_map_to_py_dict_bytes()	289
9.16.1.48 test_cpp_std_map_to_py_dict_string()	289
9.16.1.49 test_cpp_std_map_to_py_dict_string16()	289
9.16.1.50 test_cpp_std_map_to_py_dict_string32()	289
9.16.1.51 test_cpp_std_unordered_map_to_py_dict_bytes()	289
9.16.1.52 test_cpp_std_unordered_map_to_py_dict_string()	289
9.16.1.53 test_cpp_std_unordered_map_to_py_dict_string16()	290
9.16.1.54 test_cpp_std_unordered_map_to_py_dict_string32()	290
9.16.1.55 test_py_dict_to_cpp_std_map_bytes()	290
9.16.1.56 test_py_dict_to_cpp_std_map_like_bytes()	290
9.16.1.57 test_py_dict_to_cpp_std_map_like_string()	290
9.16.1.58 test_py_dict_to_cpp_std_map_like_string16()	291
9.16.1.59 test_py_dict_to_cpp_std_map_like_string32()	291
9.16.1.60 test_py_dict_to_cpp_std_map_string()	291
9.16.1.61 test_py_dict_to_cpp_std_map_string16()	291
9.16.1.62 test_py_dict_to_cpp_std_map_string32()	291
9.16.1.63 test_py_dict_to_cpp_std_unordered_map_bytes()	292
9.16.1.64 test_py_dict_to_cpp_std_unordered_map_string()	292
9.16.1.65 test_py_dict_to_cpp_std_unordered_map_u16string()	292
9.16.1.66 test_py_dict_to_cpp_std_unordered_map_u32string()	292
9.16.1.67 test_py_list_bytes_to_vector()	292
9.16.1.68 test_py_list_str16_to_vector()	293
9.16.1.69 test_py_list_str32_to_vector()	293
9.16.1.70 test_py_list_str_to_vector()	294
9.16.1.71 test_py_set_bytes_to_unordered_set()	294
9.16.1.72 test_py_set_string16_to_unordered_set()	294

9.16.1.73	test_py_set_string32_to_unordered_set()	294
9.16.1.74	test_py_set_string_to_unordered_set()	295
9.16.1.75	test_py_tuple_bytes_to_vector()	295
9.16.1.76	test_py_tuple_str16_to_vector()	295
9.16.1.77	test_py_tuple_str32_to_vector()	296
9.16.1.78	test_py_tuple_str_to_vector()	296
9.16.1.79	test_unordered_set_bytes_to_py_set()	296
9.16.1.80	test_unordered_set_string_to_py_set()	297
9.16.1.81	test_unordered_set_u16string_to_py_set()	297
9.16.1.82	test_unordered_set_u32string_to_py_set()	297
9.16.1.83	test_vector_string_to_py_list()	297
9.16.1.84	test_vector_string_to_py_tuple()	298
9.16.1.85	test_vector_u16string_to_py_list()	298
9.16.1.86	test_vector_u16string_to_py_tuple()	298
9.16.1.87	test_vector_u32string_to_py_list()	299
9.16.1.88	test_vector_u32string_to_py_tuple()	299
9.16.1.89	test_vector_vector_char_to_py_list()	300
9.16.1.90	test_vector_vector_char_to_py_tuple()	300
9.17	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.h	
	File Reference	301
9.17.1	Macro Definition Documentation	304
9.17.1.1	REPORT_OK_OR_FAIL	304
9.17.1.2	REPORT_TEST_OUTPUT	305
9.17.1.3	REPORT_TEST_OUTPUT_WITH_CONTAINER_TYPE_STRING_LENGTH	305
9.17.1.4	REPORT_TEST_OUTPUT_WITH_STRING_LENGTH	305
9.17.1.5	REPORT_TEST_OUTPUT_WITH_TYPE	305
9.17.1.6	RSS_SNAPSHOT	305
9.17.1.7	RSS_SNAPSHOT_REPORT	306
9.17.1.8	RSS_SNAPSHOT_WITH_CONTAINER_TYPE_AND_TYPE	306
9.17.1.9	RSS_SNAPSHOT_WITH_TYPE	306
9.17.1.10	RSS_SNAPSHOT_WITHOUT_TYPE	306
9.17.1.11	SET_RESULT_IF_PY_ERR_OCCURRED	306
9.17.1.12	TEST_FOR_PY_ERR_ON_ENTRY	307
9.17.1.13	TEST_FOR_PY_ERR_ON_EXIT	307
9.17.2	Function Documentation	307
9.17.2.1	compare_dict() [1/2]	307
9.17.2.2	compare_dict() [2/2]	308
9.17.2.3	compare_dict< std::map, std::string, std::string >()	308
9.17.2.4	compare_dict< std::map, std::u16string, std::u16string >()	309
9.17.2.5	compare_dict< std::map, std::u32string, std::u32string >()	309
9.17.2.6	compare_dict< std::unordered_map, std::string, std::string >()	309
9.17.2.7	compare_dict< std::unordered_map, std::u16string, std::u16string >()	309

9.17.2.8 <code>compare_dict< std::unordered_map, std::u32string, std::u32string >()</code>	309
9.17.2.9 <code>compare_list()</code> [1/2]	309
9.17.2.10 <code>compare_list()</code> [2/2]	310
9.17.2.11 <code>compare_list< bool >()</code>	311
9.17.2.12 <code>compare_list< double >()</code>	311
9.17.2.13 <code>compare_list< long >()</code>	311
9.17.2.14 <code>compare_list< std::complex< double > >()</code>	311
9.17.2.15 <code>compare_list< std::string >()</code>	311
9.17.2.16 <code>compare_list< std::vector< char > >()</code>	311
9.17.2.17 <code>compare_set()</code> [1/2]	312
9.17.2.18 <code>compare_set()</code> [2/2]	312
9.17.2.19 <code>compare_set< std::string >()</code>	313
9.17.2.20 <code>compare_set< std::u16string >()</code>	313
9.17.2.21 <code>compare_set< std::u32string >()</code>	313
9.17.2.22 <code>compare_set< std::vector< char > >()</code>	313
9.17.2.23 <code>compare_tuple()</code> [1/2]	313
9.17.2.24 <code>compare_tuple()</code> [2/2]	314
9.17.2.25 <code>compare_tuple< bool >()</code>	315
9.17.2.26 <code>compare_tuple< double >()</code>	315
9.17.2.27 <code>compare_tuple< long >()</code>	315
9.17.2.28 <code>compare_tuple< std::complex< double > >()</code>	315
9.17.2.29 <code>compare_tuple< std::string >()</code>	315
9.17.2.30 <code>compare_tuple< std::u16string >()</code>	315
9.17.2.31 <code>compare_tuple< std::u32string >()</code>	316
9.17.2.32 <code>compare_tuple< std::vector< char > >()</code>	316
9.17.2.33 <code>compare_tuple_or_list()</code>	316
9.17.2.34 <code>new_py_dict_bytes()</code>	317
9.17.2.35 <code>new_py_dict_string()</code>	317
9.17.2.36 <code>new_py_dict_string16()</code>	317
9.17.2.37 <code>new_py_dict_string32()</code>	318
9.17.2.38 <code>new_py_list_bytes()</code>	318
9.17.2.39 <code>new_py_list_string()</code>	318
9.17.2.40 <code>new_py_list_string16()</code>	319
9.17.2.41 <code>new_py_list_string32()</code>	319
9.17.2.42 <code>new_py_set_bytes()</code>	320
9.17.2.43 <code>new_py_set_string()</code>	320
9.17.2.44 <code>new_py_set_u16string()</code>	320
9.17.2.45 <code>new_py_set_u32string()</code>	321
9.17.2.46 <code>new_py_tuple_bytes()</code>	321
9.17.2.47 <code>new_py_tuple_string()</code>	321
9.17.2.48 <code>new_py_tuple_string16()</code>	322
9.17.2.49 <code>new_py_tuple_string32()</code>	322

9.17.2.50 test_cpp_std_map_like_to_py_dict()	323
9.17.2.51 test_cpp_std_map_to_py_dict()	323
9.17.2.52 test_cpp_std_map_to_py_dict_bytes()	323
9.17.2.53 test_cpp_std_map_to_py_dict_string()	323
9.17.2.54 test_cpp_std_map_to_py_dict_string16()	323
9.17.2.55 test_cpp_std_map_to_py_dict_string32()	324
9.17.2.56 test_cpp_std_unordered_map_to_py_dict()	324
9.17.2.57 test_cpp_std_unordered_map_to_py_dict_bytes()	324
9.17.2.58 test_cpp_std_unordered_map_to_py_dict_string()	324
9.17.2.59 test_cpp_std_unordered_map_to_py_dict_string16()	324
9.17.2.60 test_cpp_std_unordered_map_to_py_dict_string32()	325
9.17.2.61 test_py_dict_to_cpp_std_map()	325
9.17.2.62 test_py_dict_to_cpp_std_map_bytes()	325
9.17.2.63 test_py_dict_to_cpp_std_map_like()	325
9.17.2.64 test_py_dict_to_cpp_std_map_string()	325
9.17.2.65 test_py_dict_to_cpp_std_map_string16()	326
9.17.2.66 test_py_dict_to_cpp_std_map_string32()	326
9.17.2.67 test_py_dict_to_cpp_std_unordered_map()	326
9.17.2.68 test_py_dict_to_cpp_std_unordered_map_bytes()	326
9.17.2.69 test_py_dict_to_cpp_std_unordered_map_string()	326
9.17.2.70 test_py_dict_to_cpp_std_unordered_map_u16string()	327
9.17.2.71 test_py_dict_to_cpp_std_unordered_map_u32string()	327
9.17.2.72 test_py_list_bytes_to_vector()	327
9.17.2.73 test_py_list_str16_to_vector()	327
9.17.2.74 test_py_list_str32_to_vector()	328
9.17.2.75 test_py_list_str_to_vector()	328
9.17.2.76 test_py_list_to_vector()	329
9.17.2.77 test_py_list_to_vector_round_trip()	329
9.17.2.78 test_py_set_bytes_to_unordered_set()	329
9.17.2.79 test_py_set_string16_to_unordered_set()	329
9.17.2.80 test_py_set_string32_to_unordered_set()	329
9.17.2.81 test_py_set_string_to_unordered_set()	330
9.17.2.82 test_py_set_to_unordered_set()	330
9.17.2.83 test_py_tuple_bytes_to_vector()	330
9.17.2.84 test_py_tuple_str16_to_vector()	330
9.17.2.85 test_py_tuple_str32_to_vector()	331
9.17.2.86 test_py_tuple_str_to_vector()	331
9.17.2.87 test_py_tuple_to_vector()	332
9.17.2.88 test_py_tuple_to_vector_round_trip()	332
9.17.2.89 test_unordered_set_bytes_to_py_set()	333
9.17.2.90 test_unordered_set_string_to_py_set()	333
9.17.2.91 test_unordered_set_to_py_set()	333

9.17.2.92 test_unordered_set_u16string_to_py_set()	333
9.17.2.93 test_unordered_set_u32string_to_py_set()	334
9.17.2.94 test_vector_string_to_py_list()	334
9.17.2.95 test_vector_string_to_py_tuple()	334
9.17.2.96 test_vector_to_py_list()	335
9.17.2.97 test_vector_to_py_list_round_trip()	335
9.17.2.98 test_vector_to_py_tuple()	335
9.17.2.99 test_vector_to_py_tuple_round_trip()	336
9.17.2.100 test_vector_u16string_to_py_list()	336
9.17.2.101 test_vector_u16string_to_py_tuple()	337
9.17.2.102 test_vector_u32string_to_py_list()	337
9.17.2.103 test_vector_u32string_to_py_tuple()	338
9.17.2.104 test_vector_vector_char_to_py_list()	338
9.17.2.105 test_vector_vector_char_to_py_tuple()	339
9.17.3 Variable Documentation	339
9.17.3.1 PY_ERR_ON_ENTRY_RETURN_CODE	339
9.17.3.2 PY_ERR_ON_EXIT_RETURN_CODE	339
9.18 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.cpp	
File Reference	339
9.18.1 Function Documentation	340
9.18.1.1 test_example_cpp_std_map_to_py_dict()	340
9.18.1.2 test_example_cpp_std_unordered_map_to_py_dict()	340
9.18.1.3 test_example_py_dict_to_cpp_std_unordered_map()	340
9.18.1.4 test_example_py_tuple_to_vector_double()	340
9.18.1.5 test_example_vector_to_py_tuple_double()	341
9.18.1.6 test_functional_all()	341
9.18.1.7 test_functional_dict_copy()	341
9.18.1.8 test_functional_dict_setitem()	341
9.18.1.9 test_functional_dict_with_std_map()	341
9.18.1.10 test_functional_dict_with_std_unordred_map()	342
9.18.1.11 test_functional_frozenset_add()	342
9.18.1.12 test_functional_frozenset_add_from_iterable()	342
9.18.1.13 test_functional_list()	342
9.18.1.14 test_functional_list_setitem()	342
9.18.1.15 test_functional_set()	343
9.18.1.16 test_functional_set_add()	343
9.18.1.17 test_functional_set_add_from_iterable()	343
9.18.1.18 test_functional_tuple()	343
9.18.1.19 test_functional_tuple_setitem()	344
9.19 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.h	
File Reference	344
9.19.1 Function Documentation	344
9.19.1.1 test_functional_all()	344

9.20	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.cpp	
	File Reference	344
9.20.1	Function Documentation	345
9.20.1.1	doubles_cmp()	345
9.20.1.2	test_internal_all()	345
9.20.1.3	test_internal_test_result_atomic_test_mean_exec_time()	345
9.20.1.4	test_internal_test_result_exec_time()	346
9.20.1.5	test_internal_test_result_exec_time_min_max()	346
9.20.1.6	test_internal_test_result_string()	346
9.20.1.7	test_internal_test_result_string_multiple_a()	346
9.20.1.8	test_internal_test_result_string_multiple_b()	346
9.20.1.9	test_internal_test_result_string_using_rate()	346
9.20.1.10	test_internal_test_result_test_count()	347
9.20.1.11	test_internal_test_result_total_time()	347
9.21	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.h	
	File Reference	347
9.21.1	Function Documentation	347
9.21.1.1	test_internal_all()	347
9.22	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.cpp	
	File Reference	348
9.22.1	Function Documentation	348
9.22.1.1	test_memory_all()	348
9.22.1.2	test_memory_py_dict()	349
9.22.1.3	test_memory_py_tuple_float()	349
9.22.1.4	test_memory_py_tuple_str16_to_vector()	349
9.22.1.5	test_memory_py_tuple_str32_to_vector()	349
9.22.1.6	test_memory_py_tuple_unicode8_to_vector()	349
9.22.1.7	test_memory_py_tuple_vector_char_to_vector()	350
9.22.1.8	test_memory_test_vector_string_to_py_tuple()	350
9.22.1.9	test_memory_vector_u16string_to_py_tuple()	350
9.22.1.10	test_memory_vector_u32string_to_py_tuple()	350
9.22.1.11	test_memory_vector_vector_char_to_py_set()	350
9.22.1.12	test_memory_vector_vector_char_to_py_set_special()	351
9.22.1.13	test_memory_vector_vector_char_to_py_tuple()	351
9.23	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.h	
	File Reference	351
9.23.1	Function Documentation	351
9.23.1.1	test_memory_all()	351
9.24	/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.cpp	
	File Reference	352
9.24.1	Macro Definition Documentation	357
9.24.1.1	TEST_PERFORMANCE_DICTS	358
9.24.1.2	TEST_PERFORMANCE_FUNDAMENTAL_TYPES	358

9.24.1.3 TEST_PERFORMANCE_LISTS	358
9.24.1.4 TEST_PERFORMANCE_OBJECT_BOOL	358
9.24.1.5 TEST_PERFORMANCE_OBJECT_BYTES	358
9.24.1.6 TEST_PERFORMANCE_OBJECT_COMPLEX	358
9.24.1.7 TEST_PERFORMANCE_OBJECT_DOUBLE	358
9.24.1.8 TEST_PERFORMANCE_OBJECT_LONG	358
9.24.1.9 TEST_PERFORMANCE_OBJECT_STRING	359
9.24.1.10 TEST_PERFORMANCE_OBJECT_STRING_16	359
9.24.1.11 TEST_PERFORMANCE_OBJECT_STRING_32	359
9.24.1.12 TEST_PERFORMANCE_SETS	359
9.24.1.13 TEST_PERFORMANCE_TUPLES	359
9.24.2 Function Documentation	359
9.24.2.1 test_bool_to_py_bool_multiple()	359
9.24.2.2 test_complex_to_py_complex_multiple()	360
9.24.2.3 test_cpp_std_map_like_to_py_dict_multiple()	360
9.24.2.4 test_cpp_std_map_like_to_py_dict_string_multiple()	360
9.24.2.5 test_cpp_std_map_like_to_py_dict_vector_char_multiple()	360
9.24.2.6 test_cpp_std_map_to_py_dict_multiple()	360
9.24.2.7 test_cpp_std_map_to_py_dict_string_multiple()	361
9.24.2.8 test_cpp_std_map_to_py_dict_vector_char_multiple()	361
9.24.2.9 test_cpp_std_unordered_map_to_py_dict_multiple()	361
9.24.2.10 test_cpp_std_unordered_map_to_py_dict_string_multiple()	361
9.24.2.11 test_cpp_std_unordered_map_to_py_dict_vector_char_multiple()	361
9.24.2.12 test_cpp_string_to_py_str_multiple()	362
9.24.2.13 test_cpp_u16string_to_py_str16_multiple()	362
9.24.2.14 test_cpp_u32string_to_py_str32_multiple()	362
9.24.2.15 test_cpp_vector_char_to_py_bytes_multiple()	362
9.24.2.16 test_double_to_py_float_multiple()	363
9.24.2.17 test_list_like_string_to_py_list_multiple()	363
9.24.2.18 test_list_like_string_to_py_tuple_multiple()	363
9.24.2.19 test_list_like_to_py_list_multiple()	363
9.24.2.20 test_list_like_to_py_tuple_multiple()	364
9.24.2.21 test_list_like_u16string_to_py_list_multiple()	364
9.24.2.22 test_list_like_u32string_to_py_list_multiple()	364
9.24.2.23 test_list_like_vector_char_to_py_list_multiple()	364
9.24.2.24 test_list_like_vector_char_to_py_tuple_multiple()	365
9.24.2.25 test_list_string_to_py_list_multiple()	365
9.24.2.26 test_list_string_to_py_tuple_multiple()	365
9.24.2.27 test_list_to_py_list_multiple()	365
9.24.2.28 test_list_to_py_tuple_multiple()	365
9.24.2.29 test_list_u16string_to_py_list_multiple()	366
9.24.2.30 test_list_u32string_to_py_list_multiple()	366

9.24.2.31 test_list_vector_char_to_py_list_multiple()	366
9.24.2.32 test_list_vector_char_to_py_tuple_multiple()	366
9.24.2.33 test_long_to_py_int_multiple()	366
9.24.2.34 test_perf_cpp_std_map_to_py_dict_multiple()	367
9.24.2.35 test_perf_cpp_std_map_to_py_dict_string_multiple()	367
9.24.2.36 test_perf_cpp_std_map_to_py_dict_vector_char_multiple()	367
9.24.2.37 test_perf_cpp_std_unordered_map_to_py_dict_multiple()	367
9.24.2.38 test_perf_cpp_std_unordered_map_to_py_dict_string_multiple()	367
9.24.2.39 test_perf_cpp_std_unordered_map_to_py_dict_vector_char_multiple()	367
9.24.2.40 test_perf_list_string_to_py_list_multiple()	368
9.24.2.41 test_perf_list_string_to_py_tuple_multiple()	368
9.24.2.42 test_perf_list_to_py_list_multiple()	368
9.24.2.43 test_perf_list_to_py_tuple_multiple()	368
9.24.2.44 test_perf_list_u16string_to_py_list_multiple()	368
9.24.2.45 test_perf_list_u32string_to_py_list_multiple()	368
9.24.2.46 test_perf_list_vector_char_to_py_list_multiple()	369
9.24.2.47 test_perf_list_vector_char_to_py_tuple_multiple()	369
9.24.2.48 test_perf_py_dict_to_cpp_std_map_multiple()	369
9.24.2.49 test_perf_py_dict_to_cpp_std_map_string_multiple()	369
9.24.2.50 test_perf_py_dict_to_cpp_std_map_vector_char_multiple()	369
9.24.2.51 test_perf_py_dict_to_cpp_std_unordered_map_multiple()	369
9.24.2.52 test_perf_py_dict_to_cpp_std_unordered_map_string_multiple()	370
9.24.2.53 test_perf_py_dict_to_cpp_std_unordered_map_vector_char_multiple()	370
9.24.2.54 test_perf_py_list_to_list_multiple()	370
9.24.2.55 test_perf_py_list_to_list_string_multiple()	370
9.24.2.56 test_perf_py_list_to_list_u16string_multiple()	370
9.24.2.57 test_perf_py_list_to_list_u32string_multiple()	370
9.24.2.58 test_perf_py_list_to_list_vector_char_multiple()	371
9.24.2.59 test_perf_py_list_to_vector_multiple()	371
9.24.2.60 test_perf_py_list_to_vector_string_multiple()	371
9.24.2.61 test_perf_py_list_to_vector_u16string_multiple()	371
9.24.2.62 test_perf_py_list_to_vector_u32string_multiple()	371
9.24.2.63 test_perf_py_list_to_vector_vector_char_multiple()	371
9.24.2.64 test_perf_py_set_bytes_to_unordered_set_vector_char_multiple()	372
9.24.2.65 test_perf_py_set_str16_to_unordered_set_u16string_multiple()	372
9.24.2.66 test_perf_py_set_str32_to_unordered_set_u32string_multiple()	372
9.24.2.67 test_perf_py_set_str_to_unordered_set_string_multiple()	372
9.24.2.68 test_perf_py_set_to_unordered_set_multiple()	372
9.24.2.69 test_perf_py_tuple_to_list_multiple()	373
9.24.2.70 test_perf_py_tuple_to_list_string_multiple()	373
9.24.2.71 test_perf_py_tuple_to_list_vector_char_multiple()	373
9.24.2.72 test_perf_py_tuple_to_vector_multiple()	373

9.24.2.73 test_perf_py_tuple_to_vector_string_multiple()	373
9.24.2.74 test_perf_py_tuple_to_vector_vector_char_multiple()	373
9.24.2.75 test_perf_unordered_set_string_to_py_set_multiple()	374
9.24.2.76 test_perf_unordered_set_to_py_set_multiple()	374
9.24.2.77 test_perf_unordered_set_u16string_to_py_set_multiple()	374
9.24.2.78 test_perf_unordered_set_u32string_to_py_set_multiple()	374
9.24.2.79 test_perf_unordered_set_vector_char_to_py_set_multiple()	374
9.24.2.80 test_perf_vector_string_to_py_list_multiple()	374
9.24.2.81 test_perf_vector_string_to_py_tuple_multiple()	375
9.24.2.82 test_perf_vector_to_py_list_multiple()	375
9.24.2.83 test_perf_vector_to_py_tuple_multiple()	375
9.24.2.84 test_perf_vector_u16string_to_py_list_multiple()	375
9.24.2.85 test_perf_vector_u32string_to_py_list_multiple()	375
9.24.2.86 test_perf_vector_vector_char_to_py_list_multiple()	375
9.24.2.87 test_perf_vector_vector_char_to_py_tuple_multiple()	376
9.24.2.88 test_performance_all()	376
9.24.2.89 test_py_bool_to_cpp_bool_multiple()	376
9.24.2.90 test_py_bytes_to_cpp_vector_char_multiple()	376
9.24.2.91 test_py_complex_to_cpp_complex_multiple()	376
9.24.2.92 test_py_dict_to_cpp_std_map_like_multiple()	376
9.24.2.93 test_py_dict_to_cpp_std_map_like_string_multiple()	377
9.24.2.94 test_py_dict_to_cpp_std_map_like_vector_char_multiple()	377
9.24.2.95 test_py_dict_to_cpp_std_map_multiple()	377
9.24.2.96 test_py_dict_to_cpp_std_map_string_multiple()	377
9.24.2.97 test_py_dict_to_cpp_std_map_vector_char_multiple()	377
9.24.2.98 test_py_dict_to_cpp_std_unordered_map_multiple()	378
9.24.2.99 test_py_dict_to_cpp_std_unordered_map_string_multiple()	378
9.24.2.100 test_py_dict_to_cpp_std_unordered_map_vector_char_multiple()	378
9.24.2.101 test_py_float_to_cpp_double_multiple()	378
9.24.2.102 test_py_int_to_cpp_long_multiple()	378
9.24.2.103 test_py_list_bytes_to_list_like_vector_char_multiple()	379
9.24.2.104 test_py_list_bytes_to_list_vector_char_multiple()	379
9.24.2.105 test_py_list_bytes_to_vector_vector_char_multiple()	379
9.24.2.106 test_py_list_str16_to_list_like_u16string_multiple()	379
9.24.2.107 test_py_list_str16_to_list_u16string_multiple()	379
9.24.2.108 test_py_list_str16_to_vector_u16string_multiple()	380
9.24.2.109 test_py_list_str32_to_list_like_u32string_multiple()	380
9.24.2.110 test_py_list_str32_to_list_u32string_multiple()	380
9.24.2.111 test_py_list_str32_to_vector_u32string_multiple()	380
9.24.2.112 test_py_list_str_to_list_like_string_multiple()	380
9.24.2.113 test_py_list_str_to_list_string_multiple()	381
9.24.2.114 test_py_list_str_to_vector_string_multiple()	381

9.24.2.115 test_py_list_to_list_like_multiple()	381
9.24.2.116 test_py_list_to_list_multiple()	381
9.24.2.117 test_py_list_to_vector_multiple()	381
9.24.2.118 test_py_set_bytes_to_unordered_set_vector_char_multiple()	382
9.24.2.119 test_py_set_str16_to_unordered_set_u16string_multiple()	382
9.24.2.120 test_py_set_str32_to_unordered_set_u32string_multiple()	382
9.24.2.121 test_py_set_str_to_unordered_set_string_multiple()	382
9.24.2.122 test_py_set_to_unordered_set_multiple()	382
9.24.2.123 test_py_str16_to_cpp_u16string_multiple()	383
9.24.2.124 test_py_str32_to_cpp_u32string_multiple()	383
9.24.2.125 test_py_str_to_cpp_string_multiple()	383
9.24.2.126 test_py_tuple_bytes_to_list_like_vector_char_multiple()	383
9.24.2.127 test_py_tuple_bytes_to_list_vector_char_multiple()	383
9.24.2.128 test_py_tuple_bytes_to_vector_vector_char_multiple()	384
9.24.2.129 test_py_tuple_str_to_list_like_string_multiple()	384
9.24.2.130 test_py_tuple_str_to_list_string_multiple()	384
9.24.2.131 test_py_tuple_str_to_vector_string_multiple()	384
9.24.2.132 test_py_tuple_to_list_like_multiple()	384
9.24.2.133 test_py_tuple_to_list_multiple()	385
9.24.2.134 test_py_tuple_to_vector_multiple()	385
9.24.2.135 test_unordered_set_string_to_py_set_multiple()	385
9.24.2.136 test_unordered_set_to_py_set_multiple()	385
9.24.2.137 test_unordered_set_u16string_to_py_set_multiple()	385
9.24.2.138 test_unordered_set_u32string_to_py_set_multiple()	386
9.24.2.139 test_unordered_set_vector_char_to_py_set_multiple()	386
9.24.2.140 test_vector_string_to_py_list_multiple()	386
9.24.2.141 test_vector_string_to_py_tuple_multiple()	386
9.24.2.142 test_vector_to_py_list_multiple()	386
9.24.2.143 test_vector_to_py_tuple_multiple()	387
9.24.2.144 test_vector_u16string_to_py_list_multiple()	387
9.24.2.145 test_vector_u32string_to_py_list_multiple()	387
9.24.2.146 test_vector_vector_char_to_py_list_multiple()	387
9.24.2.147 test_vector_vector_char_to_py_tuple_multiple()	387
9.24.3 Variable Documentation	388
9.24.3.1 INC_SIZE_OF_CONTAINER_MULTIPLE	388
9.24.3.2 INC_STRING_LENGTH_MULTIPLE	388
9.24.3.3 LIMIT_SIZE_OF_CONTAINER	388
9.24.3.4 LIMIT_SIZE_OF_CONTAINER_DICT	388
9.24.3.5 LIMIT_STRING_LENGTH	388
9.24.3.6 MIN_SIZE_OF_CONTAINER	388
9.24.3.7 MIN_STRING_LENGTH_HASHABLE	389
9.24.3.8 MIN_STRING_LENGTH_NON_HASHABLE	389

9.24.3.9 TEST_REPEAT	389
9.25 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.h	
File Reference	389
9.25.1 Function Documentation	389
9.25.1.1 test_performance_all()	389
9.26 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cPyCppContainers.cpp	
File Reference	389
9.26.1 Macro Definition Documentation	391
9.26.1.1 PY_SSIZE_T_CLEAN	391
9.26.1.2 SINGLE_ARGUMENT_METHOD	392
9.26.2 Function Documentation	392
9.26.2.1 dict_inc()	392
9.26.2.2 list_x2()	392
9.26.2.3 new_bytes()	393
9.26.2.4 new_dict()	393
9.26.2.5 new_dict_debug()	393
9.26.2.6 new_dict_debug_float_float()	393
9.26.2.7 new_dict_debug_int_int()	394
9.26.2.8 new_dict_from_std_map_bytes_bytes()	394
9.26.2.9 new_dict_from_std_map_complex_complex()	394
9.26.2.10 new_dict_from_std_map_float_float()	394
9.26.2.11 new_dict_from_std_map_int_int()	394
9.26.2.12 new_dict_from_std_map_int_str()	394
9.26.2.13 new_dict_from_std_map_int_str16()	395
9.26.2.14 new_dict_from_std_map_int_str32()	395
9.26.2.15 new_dict_from_std_map_str16_str16()	395
9.26.2.16 new_dict_from_std_map_str32_str32()	395
9.26.2.17 new_dict_from_std_map_str_str()	395
9.26.2.18 new_dict_from_std_unordered_map_bytes_bytes()	395
9.26.2.19 new_dict_from_std_unordered_map_complex_complex()	396
9.26.2.20 new_dict_from_std_unordered_map_float_float()	396
9.26.2.21 new_dict_from_std_unordered_map_int_int()	396
9.26.2.22 new_dict_from_std_unordered_map_int_str()	396
9.26.2.23 new_dict_from_std_unordered_map_int_str16()	396
9.26.2.24 new_dict_from_std_unordered_map_int_str32()	396
9.26.2.25 new_dict_from_std_unordered_map_str16_str16()	397
9.26.2.26 new_dict_from_std_unordered_map_str32_str32()	397
9.26.2.27 new_dict_from_std_unordered_map_str_str()	397
9.26.2.28 new_frozenset()	397
9.26.2.29 new_frozenset_bytes()	397
9.26.2.30 new_frozenset_complex()	398
9.26.2.31 new_frozenset_float()	398
9.26.2.32 new_frozenset_int()	399

9.26.2.33 new_frozenset_str()	399
9.26.2.34 new_list()	399
9.26.2.35 new_list_list_bool()	400
9.26.2.36 new_list_list_bytes()	400
9.26.2.37 new_list_list_complex()	400
9.26.2.38 new_list_list_float()	400
9.26.2.39 new_list_list_int()	400
9.26.2.40 new_list_list_str()	401
9.26.2.41 new_list_list_str16()	401
9.26.2.42 new_list_list_str32()	401
9.26.2.43 new_list_vector_bool()	401
9.26.2.44 new_list_vector_bytes()	401
9.26.2.45 new_list_vector_complex()	402
9.26.2.46 new_list_vector_float()	402
9.26.2.47 new_list_vector_int()	402
9.26.2.48 new_list_vector_str()	402
9.26.2.49 new_list_vector_str16()	402
9.26.2.50 new_list_vector_str32()	402
9.26.2.51 new_set()	402
9.26.2.52 new_set_bytes()	403
9.26.2.53 new_set_complex()	403
9.26.2.54 new_set_float()	404
9.26.2.55 new_set_int()	404
9.26.2.56 new_set_str()	404
9.26.2.57 new_set_str16()	405
9.26.2.58 new_set_str32()	405
9.26.2.59 new_str()	405
9.26.2.60 new_str16()	406
9.26.2.61 new_str32()	406
9.26.2.62 PyInit_cPyCppContainers()	406
9.26.2.63 reverse_vector()	406
9.26.2.64 tuple_reverse()	407
9.26.2.65 vector_double_x2()	407
9.26.3 Variable Documentation	407
9.26.3.1 cPyCppContainersMethods	407
9.26.3.2 cPyCppContainersmodule	407
9.27 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.cpp File Reference	408
9.27.1 Macro Definition Documentation	409
9.27.1.1 PY_SSIZE_T_CLEAN	409
9.27.2 Function Documentation	409
9.27.2.1 cpp_custom_object_to_py_custom_object()	409

9.27.2.2 Custom_dealloc()	409
9.27.2.3 Custom_getfirst()	410
9.27.2.4 Custom_getlast()	410
9.27.2.5 Custom_init()	410
9.27.2.6 Custom_name()	410
9.27.2.7 Custom_new()	410
9.27.2.8 Custom_setfirst()	410
9.27.2.9 Custom_setlast()	411
9.27.2.10 py_custom_object_check()	411
9.27.2.11 py_custom_object_to_cpp_custom_object()	411
9.27.2.12 PyInit_cUserDefined()	411
9.27.2.13 reverse_dict_names()	411
9.27.2.14 reverse_list_names()	411
9.27.3 Variable Documentation	411
9.27.3.1 cUserDefinedMethods	412
9.27.3.2 cUserDefinedmodule	412
9.27.3.3 Custom_getsetters	412
9.27.3.4 Custom_members	412
9.27.3.5 Custom_methods	413
9.27.3.6 CustomType	413
9.28 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.h File Reference	413
9.29 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/custom_3_Python3.9.0.c File Reference	414
9.30 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/main.cpp File Reference	414
9.30.1 Macro Definition Documentation	414
9.30.1.1 TEST_FUNCTIONAL_ALL	415
9.30.1.2 TEST_INTERNAL_ALL	415
9.30.1.3 TEST_MEMORY_ALL	415
9.30.1.4 TEST_PERFORMANCE_ALL	415
9.30.2 Function Documentation	415
9.30.2.1 explore_hash_reserve()	415
9.30.2.2 main()	415
9.30.2.3 test_all()	415
9.31 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen.py File Reference	416
9.32 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_common.py File Reference	417
9.33 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_documentation.py File Reference	417
9.34 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ReadMe.md File Reference	418
9.35 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/README.md File Reference	418

Chapter 1

Python and C++ Containers

Python is well known for its ability to handle *heterogeneous* data in containers such as lists. But what if you need to convert to and from C++ containers such as `std::vector<T>` that require *homogeneous* data types?

This C++ project is about converting between C++ containers and Python's (tuple, list, set, frozenset, dict) containing homogeneous types (bool, int, float, complex, bytes, str) to and from their C++ equivalents.

These type objects are supported:

C++ Type	Python Equivalent
bool	True, False
long	int
double	float
<code>std::complex<double></code>	complex
<code>std::vector<char></code>	bytes
<code>std::string</code>	str
<code>std::u16string</code>	str
<code>std::u32string</code>	str

With two-way conversion for this set of containers:

C++ Container	Python Equivalent
<code>std::vector<T></code>	list
<code>std::vector<T></code>	tuple
<code>std::list<T></code>	list
<code>std::list<T></code>	tuple
<code>std::unordered_set<T></code>	set
<code>std::unordered_set<T></code>	frozenset
<code>std::unordered_map<K, V></code>	dict
<code>std::map<K, V></code>	dict

These combinations would normally need 352 specific conversion functions.

This project reduces that to just **six** hand maintained functions. The 352 actual conversion functions are then created automatically using a mixture of templates, partial specialisation and code generation. This approach means that new types and containers can be added with ease.

1.1 Using This Library

1.1.1 C++ To Python

Suppose that you have a Python list of floats that needs to be passed to a C++ function that expects `std::vector<double>`. Then that C++ function modifies that vector and you need the result as a new Python list of floats. With this library your code will be as simple as this:

```
{c++}
#include "python_convert.h"
static PyObject *
your_function_name(void) {
    std::vector<double> container = some_cpp_function_that_creates_a_vector();
    // Convert the vector back to a new Python list of float
    // with a generic function.
    return Python_Cpp_Containers::cpp_std_list_like_to_py_list(container);
}
```

Some other variations, firstly create a Python tuple rather than a list:

```
{c++}
#include "python_convert.h"
static PyObject *
your_function_name(void) {
    std::vector<double> container = some_cpp_function_that_creates_a_vector();
    return Python_Cpp_Containers::cpp_std_list_like_to_py_tuple(container);
}
```

Or work with a `std::list` rather than a `std::vector`:

```
{c++}
#include "python_convert.h"
static PyObject *
your_function_name(void) {
    std::list<double> container = some_cpp_function_that_creates_a_list();
    return Python_Cpp_Containers::cpp_std_list_like_to_py_list(container);
}
```

Or work with a `std::vector<std::string>`:

```
{c++}
#include "python_convert.h"
static PyObject *
your_function_name(void) {
    std::vector<std::string> container = some_cpp_function_that_creates_a_vector();
    return Python_Cpp_Containers::cpp_std_list_like_to_py_list(container);
}
```

Note `Python_Cpp_Containers::cpp_std_list_like_to_py_list(container)` will select the correct type conversion or will give a compile time error if there is a type mismatch.

1.1.2 Python to C++

```
{c++}
#include "python_convert.h"
static PyObject *
your_function_name(PyObject *arg) {
    // Declare the specific vector type
    std::vector<double> vec;
    // Call the generic function to convert a list to a std::vector.
    // This returns non-zero if it can not convert arg to a
    // std::vector<double>
    if (!Python_Cpp_Containers::py_list_to_cpp_std_list_like(arg, vec)) {
        // Send the std::vector<double> to the C++ library.
        // ...
        Py_RETURN_NONE;
    }
    return NULL;
}
```

1.2 Usage

1.2.1 Code Generation

If necessary run the code generator:

```
cd src/py
python code_gen.py
```

Which should give you something like:

```
venv/bin/python src/py/code_gen.py
Version: 0.4.0
Target directory "src/cpy"
Writing declarations to "src/cpy/auto_py_convert_internal.h"
Wrote 4125 lines of code with 356 declarations.
Writing definitions to "src/cpy/auto_py_convert_internal.cpp"
Wrote 3971 lines of code with 352 definitions.
Process finished with exit code 0
```

1.2.2 C++ Build Configuration

You need to compile the following C++ files by adding them to your makefile or CMakeLists.txt:

```
src/cpy/auto_py_convert_internal.cpp
src/cpy/python_container_convert.cpp
src/cpy/python_object_convert.cpp
```

1.2.2.1 Source Inclusion

Your pre-processor needs access to the header files with the compiler flag `-I src/cpy`.

Then in your C++ code include:

```
{c++}
#include "python_convert.h"
```

Which gives you access to the whole API in the namespace `Python_Cpp_Containers`.

1.2.3 Python Extension Example

There are some examples of using this library in [src/ext/cPyCppContainers.cpp](#). This extension is built by `setup.py` and tested with `tests/unit/test_cPyCppContainers.py`.

To build this extension:

```
$ python setup.py develop
```

And to use it:

```
import cPyCppContainers
```

There are a number of functions there that exploit this C++ library. For example this C function create a C++ `std::vector<double>` from a Python list of floats then creates a new Python list of floats from that C++ container ('round-tripping').

```
{c++}
static PyObject *
new_list_float(PyObject *arg) {
    std::vector<double> vec;
    if (!py_list_to_cpp_std_list_like(arg, vec)) {
        return cpp_std_list_like_to_py_list(vec);
    }
    return NULL;
}
```

This can be called from Python thus:

```
import cPyCppContainers
cPyCppContainers.new_list_float([1.0, 2.0])
[1.0, 2.0]
```

If the Python list contains non-floats an exception will be raised:

```
import cPyCppContainers
cPyCppContainers.new_list_float([1.0, 2])
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: Python value of type int can not be converted
```

1.2.3.1 Testing

To test the cPyCppContainers extension which exercises much of the C++ code:

```
$ pytest tests/
```

This takes a few seconds. There are a couple of options that can be added:

- `--runslow` will run slow tests including performance test. Use the `-s` option to obtain the performance output.
- `--pymemtrace` will run memory tracing tests. This requires `pymemtrace` to be installed.

So for the full set of tests:

```
$ pytest -vs --runslow --pymemtrace tests/
```

This can take around 30 minutes to complete.

Chapter 2

Source Code for Python Cpp Homogenous Containers

In this `src/` directory are these subdirectories:

- `cpp/` Pure C++ code, no Python API is used.
- `cpy/` C/C++ code that `#includes Python.h`.
- `ext/` Python extensions in C/C++ using Python APIs by `#including Python.h`.
- `py/` Pure Python code.

2.1 `cpp/` Files of Note

- `get_rss.h` and `get_rss.cpp`: Code to get the Resident Set Size (RSS).
- `save_stream_state.h`: Saves and restores the state of a stream.
- `TestFramework.h` and `TestFramework.cpp`: C++ code that creates a test framework.

2.2 `cpy/` Files and Directories of Note

In the `test/` directory are all the C++ test files including functional and performance tests.

`auto_py_convert_internal.h` and `auto_py_convert_internal.cpp` is the C++ code that is generated by `code_gen.py`. These C++ files should *not* be edited.

`python_container_convert.h` and `python_container_convert.cpp` contain the handwritten C++ code that provides functions to interact with Python containers such as tuples. Namely:

- Check if a Python container is a particular type.
- Create a new Python container.
- Find the size of the Python container.

- Set a `PyObject*` into a Python container.
- Get a `PyObject*` from a Python container.

These are all very simple functions that wrap existing Python API functions or macros.

[python_convert.h](#) contains the hand maintained C++ code that converts all combinations of Python containers and objects, such as floats, to and from their C/C++ equivalents. There are just six handwritten C++ template functions here.

[python_object_convert.h](#) and [python_object_convert.cpp](#) contain the handwritten C++ code that converts Python objects, such as floats, to and from their C/C++ equivalents. Namely:

- Check if a Python object is a particular type.
- Convert a `PyObject*` into a C++ type.
- Convert a C++ type into a `PyObject*`.

These are all very simple functions that wrap existing Python API functions or macros.

2.3 `<tt>ext/` Files of Note

[cPyCppContainers.cpp](#) provides examples of interfacing between C++ and Python with this code. This extension is used for the round trip performance tests.

2.4 `<tt>py/` Files of Note

This contains the Python scripts that generates C++ code into the `cpy/` directory. [code_gen.py](#) is the main script.

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Python_Cpp_Containers	
Conversion functions for individual Python objects	15
src	195
src.py	195
src.py.code_gen	195
src.py.code_gen_common	203
src.py.code_gen_documentation	203
std	208

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

CppCustomObject	209
CustomObject	211
ExecClock	212
std::hash< std::complex< double > >	213
std::hash< std::vector< char > >	213
std::less< std::complex< T > >	214
typing.NamedTuple	
src.py.code_gen.CodeCount	209
src.py.code_gen_common.CppTypeFunctions	211
src.py.code_gen_common.TypeConversionFunctions	229
src.py.code_gen_common.UnaryFunctions	229
RSSSnapshot	214
StreamFormatState	218
SubTestCount	219
TestResult	220
TestResultS	227

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

src.py.code_gen.CodeCount	209
CppCustomObject	209
src.py.code_gen_common.CppTypeFunctions	211
CustomObject	211
ExecClock	212
std::hash< std::complex< double > >	213
std::hash< std::vector< char > >	213
std::less< std::complex< T > >	214
RSSSnapshot	214
StreamFormatState	218
SubTestCount	219
TestResult	220
TestResultS	227
src.py.code_gen_common.TypeConversionFunctions	229
src.py.code_gen_common.UnaryFunctions	229

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/___init__.py	231
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/main.cpp	414
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/get_rss.cpp	231
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/get_rss.h	234
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/save_stream_state.h	235
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.cpp	236
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.h	241
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/auto_py_convert_internal.cpp 244	
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/auto_py_convert_internal.h	256
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_container_convert.cpp 268	
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_container_convert.h	268
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert.h	269
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert_scrap.h	271
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_convert.cpp	273
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_convert.h	274
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.cpp	275
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.h	301
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.cpp	339
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.h	344
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.cpp	344
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.h	347
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.cpp	348
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.h	351
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.cpp	352
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.h	389
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cPyCppContainers.cpp	389
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.cpp	408
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.h	413
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/custom_3_Python3.9.0.c	414
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/___init__.py	231
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen.py	416
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_common.py	417
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_documentation.py	417

Chapter 7

Namespace Documentation

7.1 Python_Cpp_Containers Namespace Reference

Conversion functions for individual Python objects.

Enumerations

- enum class [ErrorReturnValue](#) : int {
 [SUCCESS](#) = 0 , [FAIL_CONTAINER_WRONG_TYPE](#) , [FAIL_CONTAINER_MEMBER_WRONG_TYPE](#) ,
 [FAIL_CONTAINER_KEY_WRONG_TYPE](#) ,
 [FAIL_CONTAINER_VALUE_WRONG_TYPE](#) }

Functions

- template<> PyObject * [cpp_std_list_like_to_py_tuple< bool >](#) (const std::vector< bool > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< long >](#) (const std::vector< long > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< double >](#) (const std::vector< double > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< std::complex< double > >](#) (const std::vector< std::complex< double > > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< std::vector< char > >](#) (const std::vector< std::vector< char > > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< std::string >](#) (const std::vector< std::string > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< std::u16string >](#) (const std::vector< std::u16string > &container)
- template<> PyObject * [cpp_std_list_like_to_py_tuple< std::u32string >](#) (const std::vector< std::u32string > &container)
- template<> int [py_tuple_to_cpp_std_list_like< bool >](#) (PyObject *op, std::vector< bool > &container)
- template<> int [py_tuple_to_cpp_std_list_like< long >](#) (PyObject *op, std::vector< long > &container)
- template<> int [py_tuple_to_cpp_std_list_like< double >](#) (PyObject *op, std::vector< double > &container)
- template<> int [py_tuple_to_cpp_std_list_like< std::complex< double > >](#) (PyObject *op, std::vector< std::complex< double > > &container)
- template<> int [py_tuple_to_cpp_std_list_like< std::vector< char > >](#) (PyObject *op, std::vector< std::vector< char > > &container)
- template<> int [py_tuple_to_cpp_std_list_like< std::string >](#) (PyObject *op, std::vector< std::string > &container)

- `template<> int py_tuple_to_cpp_std_list_like< std::u16string > (PyObject *op, std::vector< std::u16string > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< std::u32string > (PyObject *op, std::vector< std::u32string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< bool > (const std::list< bool > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< long > (const std::list< long > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< double > (const std::list< double > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< std::complex< double > > (const std::list< std::complex< double > > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< std::vector< char > > (const std::list< std::vector< char > > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< std::string > (const std::list< std::string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< std::u16string > (const std::list< std::u16string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_tuple< std::u32string > (const std::list< std::u32string > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< bool > (PyObject *op, std::list< bool > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< double > (PyObject *op, std::list< double > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::list< std::complex< double > > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::list< std::vector< char > > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< std::string > (PyObject *op, std::list< std::string > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< std::u16string > (PyObject *op, std::list< std::u16string > &container)`
- `template<> int py_tuple_to_cpp_std_list_like< std::u32string > (PyObject *op, std::list< std::u32string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< bool > (const std::vector< bool > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< long > (const std::vector< long > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< double > (const std::vector< double > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::complex< double > > (const std::vector< std::complex< double > > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::vector< char > > (const std::vector< std::vector< char > > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::string > (const std::vector< std::string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::u16string > (const std::vector< std::u16string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::u32string > (const std::vector< std::u32string > &container)`
- `template<> int py_list_to_cpp_std_list_like< bool > (PyObject *op, std::vector< bool > &container)`
- `template<> int py_list_to_cpp_std_list_like< long > (PyObject *op, std::vector< long > &container)`
- `template<> int py_list_to_cpp_std_list_like< double > (PyObject *op, std::vector< double > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::vector< std::complex< double > > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::vector< std::vector< char > > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::string > (PyObject *op, std::vector< std::string > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::u16string > (PyObject *op, std::vector< std::u16string > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::u32string > (PyObject *op, std::vector< std::u32string > &container)`

- `template<> PyObject * cpp_std_list_like_to_py_list< bool > (const std::list< bool > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< long > (const std::list< long > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< double > (const std::list< double > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::complex< double > > (const std::list< std::complex< double > > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::vector< char > > (const std::list< std::vector< char > > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::string > (const std::list< std::string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::u16string > (const std::list< std::u16string > &container)`
- `template<> PyObject * cpp_std_list_like_to_py_list< std::u32string > (const std::list< std::u32string > &container)`
- `template<> int py_list_to_cpp_std_list_like< bool > (PyObject *op, std::list< bool > &container)`
- `template<> int py_list_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`
- `template<> int py_list_to_cpp_std_list_like< double > (PyObject *op, std::list< double > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::list< std::complex< double > > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::list< std::vector< char > > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::string > (PyObject *op, std::list< std::string > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::u16string > (PyObject *op, std::list< std::u16string > &container)`
- `template<> int py_list_to_cpp_std_list_like< std::u32string > (PyObject *op, std::list< std::u32string > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< bool > (const std::unordered_set< bool > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< long > (const std::unordered_set< long > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< double > (const std::unordered_set< double > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< std::complex< double > > (const std::unordered_set< std::complex< double > > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< std::vector< char > > (const std::unordered_set< std::vector< char > > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< std::string > (const std::unordered_set< std::string > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< std::u16string > (const std::unordered_set< std::u16string > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_set< std::u32string > (const std::unordered_set< std::u32string > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< bool > (PyObject *op, std::unordered_set< bool > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< long > (PyObject *op, std::unordered_set< long > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< double > (PyObject *op, std::unordered_set< double > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< std::complex< double > > (PyObject *op, std::unordered_set< std::complex< double > > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< std::vector< char > > (PyObject *op, std::unordered_set< std::vector< char > > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< std::string > (PyObject *op, std::unordered_set< std::string > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< std::u16string > (PyObject *op, std::unordered_set< std::u16string > &container)`
- `template<> int py_set_to_cpp_std_unordered_set< std::u32string > (PyObject *op, std::unordered_set< std::u32string > &container)`

- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< bool > (const std::unordered_set< bool > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< long > (const std::unordered_set< long > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< double > (const std::unordered_set< double > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< std::complex< double > > (const std::unordered_set< std::complex< double > > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< std::vector< char > > (const std::unordered_set< std::vector< char > > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< std::string > (const std::unordered_set< std::string > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< std::u16string > (const std::unordered_set< std::u16string > &container)`
- `template<> PyObject * cpp_std_unordered_set_to_py_frozenset< std::u32string > (const std::unordered_set< std::u32string > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< bool > (PyObject *op, std::unordered_set< bool > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< long > (PyObject *op, std::unordered_set< long > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< double > (PyObject *op, std::unordered_set< double > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< std::complex< double > > (PyObject *op, std::unordered_set< std::complex< double > > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< std::vector< char > > (PyObject *op, std::unordered_set< std::vector< char > > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< std::string > (PyObject *op, std::unordered_set< std::string > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< std::u16string > (PyObject *op, std::unordered_set< std::u16string > &container)`
- `template<> int py_frozenset_to_cpp_std_unordered_set< std::u32string > (PyObject *op, std::unordered_set< std::u32string > &container)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool > (const std::unordered_map< bool, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, bool > (PyObject *op, std::unordered_map< bool, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, long > (const std::unordered_map< bool, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, long > (PyObject *op, std::unordered_map< bool, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, double > (const std::unordered_map< bool, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, double > (PyObject *op, std::unordered_map< bool, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::complex< double > > (const std::unordered_map< bool, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::complex< double > > (PyObject *op, std::unordered_map< bool, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > > (const std::unordered_map< bool, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::vector< char > > (PyObject *op, std::unordered_map< bool, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::string > (const std::unordered_map< bool, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::string > (PyObject *op, std::unordered_map< bool, std::string > &map)`

- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string > (const std::unordered_map< bool, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u16string > (PyObject *op, std::unordered_map< bool, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string > (const std::unordered_map< bool, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u32string > (PyObject *op, std::unordered_map< bool, std::u32string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, bool > (const std::unordered_map< long, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, bool > (PyObject *op, std::unordered_map< long, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, long > (const std::unordered_map< long, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, long > (PyObject *op, std::unordered_map< long, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, double > (const std::unordered_map< long, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, double > (PyObject *op, std::unordered_map< long, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, std::complex< double > > (const std::unordered_map< long, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex< double > > (PyObject *op, std::unordered_map< long, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > > (const std::unordered_map< long, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, std::vector< char > > (PyObject *op, std::unordered_map< long, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string > (const std::unordered_map< long, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, std::string > (PyObject *op, std::unordered_map< long, std::string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string > (const std::unordered_map< long, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u16string > (PyObject *op, std::unordered_map< long, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u32string > (const std::unordered_map< long, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u32string > (PyObject *op, std::unordered_map< long, std::u32string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, double, bool > (const std::unordered_map< double, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, double, bool > (PyObject *op, std::unordered_map< double, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, double, long > (const std::unordered_map< double, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, double, long > (PyObject *op, std::unordered_map< double, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, double, double > (const std::unordered_map< double, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, double, double > (PyObject *op, std::unordered_map< double, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, double, std::complex< double > > (const std::unordered_map< double, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, double, std::complex< double > > (PyObject *op, std::unordered_map< double, std::complex< double > > &map)`

- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, double > (const std::unordered_map< std::u32string, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, double > (PyObject *op, std::unordered_map< std::u32string, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::complex< double > > (const std::unordered_map< std::u32string, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::complex< double > > (PyObject *op, std::unordered_map< std::u32string, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::vector< char > > (const std::unordered_map< std::u32string, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::vector< char > > (PyObject *op, std::unordered_map< std::u32string, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::string > (const std::unordered_map< std::u32string, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::string > (PyObject *op, std::unordered_map< std::u32string, std::string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u16string > (const std::unordered_map< std::u32string, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u16string > (PyObject *op, std::unordered_map< std::u32string, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u32string > (const std::unordered_map< std::u32string, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u32string > (PyObject *op, std::unordered_map< std::u32string, std::u32string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, bool > (const std::map< bool, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, bool > (PyObject *op, std::map< bool, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, long > (const std::map< bool, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, long > (PyObject *op, std::map< bool, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, double > (const std::map< bool, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, double > (PyObject *op, std::map< bool, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, std::complex< double > > (const std::map< bool, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, std::complex< double > > (PyObject *op, std::map< bool, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, std::vector< char > > (const std::map< bool, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, std::vector< char > > (PyObject *op, std::map< bool, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, std::string > (const std::map< bool, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, std::string > (PyObject *op, std::map< bool, std::string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, std::u16string > (const std::map< bool, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, std::u16string > (PyObject *op, std::map< bool, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, bool, std::u32string > (const std::map< bool, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, bool, std::u32string > (PyObject *op, std::map< bool, std::u32string > &map)`

- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, bool > (const std::map< long, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, bool > (PyObject *op, std::map< long, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, long > (const std::map< long, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, long > (PyObject *op, std::map< long, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, double > (const std::map< long, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, double > (PyObject *op, std::map< long, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, std::complex< double > > (const std::map< long, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, std::complex< double > > (PyObject *op, std::map< long, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, std::vector< char > > (const std::map< long, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > > (PyObject *op, std::map< long, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, std::string > (const std::map< long, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, std::string > (PyObject *op, std::map< long, std::string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, std::u16string > (const std::map< long, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, std::u16string > (PyObject *op, std::map< long, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, std::u32string > (const std::map< long, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, std::u32string > (PyObject *op, std::map< long, std::u32string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, double, bool > (const std::map< double, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, double, bool > (PyObject *op, std::map< double, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, double, long > (const std::map< double, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, double, long > (PyObject *op, std::map< double, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, double, double > (const std::map< double, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, double, double > (PyObject *op, std::map< double, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, double, std::complex< double > > (const std::map< double, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, double, std::complex< double > > (PyObject *op, std::map< double, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, double, std::vector< char > > (const std::map< double, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > > (PyObject *op, std::map< double, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, double, std::string > (const std::map< double, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, double, std::string > (PyObject *op, std::map< double, std::string > &map)`

- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::string, std::u16string > (const std::map< std::string, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string > (PyObject *op, std::map< std::string, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::string, std::u32string > (const std::map< std::string, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string > (PyObject *op, std::map< std::string, std::u32string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, bool > (const std::map< std::u16string, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, bool > (PyObject *op, std::map< std::u16string, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, long > (const std::map< std::u16string, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, long > (PyObject *op, std::map< std::u16string, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, double > (const std::map< std::u16string, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, double > (PyObject *op, std::map< std::u16string, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double > > (const std::map< std::u16string, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex< double > > (PyObject *op, std::map< std::u16string, std::complex< double > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > > (const std::map< std::u16string, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector< char > > (PyObject *op, std::map< std::u16string, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string > (const std::map< std::u16string, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string > (PyObject *op, std::map< std::u16string, std::string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string > (const std::map< std::u16string, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string > (PyObject *op, std::map< std::u16string, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string > (const std::map< std::u16string, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string > (PyObject *op, std::map< std::u16string, std::u32string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, bool > (const std::map< std::u32string, bool > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, bool > (PyObject *op, std::map< std::u32string, bool > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, long > (const std::map< std::u32string, long > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, long > (PyObject *op, std::map< std::u32string, long > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, double > (const std::map< std::u32string, double > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, double > (PyObject *op, std::map< std::u32string, double > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, std::complex< double > > (const std::map< std::u32string, std::complex< double > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex< double > > (PyObject *op, std::map< std::u32string, std::complex< double > > &map)`

- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, std::vector< char > > (const std::map< std::u32string, std::vector< char > > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector< char > > (PyObject *op, std::map< std::u32string, std::vector< char > > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, std::string > (const std::map< std::u32string, std::string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string > (PyObject *op, std::map< std::u32string, std::string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u16string > (const std::map< std::u32string, std::u16string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string > (PyObject *op, std::map< std::u32string, std::u16string > &map)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u32string > (const std::map< std::u32string, std::u32string > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string > (PyObject *op, std::map< std::u32string, std::u32string > &map)`
- `template<typename T > PyObject * cpp_std_list_like_to_py_tuple (const std::vector< T > &container)`
- `template<typename T > int py_tuple_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<typename T > PyObject * cpp_std_list_like_to_py_tuple (const std::list< T > &container)`
- `template<typename T > int py_tuple_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<typename T > PyObject * cpp_std_list_like_to_py_list (const std::vector< T > &container)`
- `template<typename T > int py_list_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<typename T > PyObject * cpp_std_list_like_to_py_list (const std::list< T > &container)`
- `template<typename T > int py_list_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<typename T > PyObject * cpp_std_unordered_set_to_py_set (const std::unordered_set< T > &container)`
- `template<typename T > int py_set_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &container)`
- `template<typename T > PyObject * cpp_std_unordered_set_to_py_frozenset (const std::unordered_set< T > &container)`
- `template<typename T > int py_frozenset_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &container)`
- `template<template< typename ... > class Map, typename K, typename V > PyObject * cpp_std_map_like_to_py_dict (const Map< K, V > &map)`
- `template<template< typename ... > class Map, typename K, typename V > int py_dict_to_cpp_std_map_like (PyObject *op, Map< K, V > &map)`
- `int py_tuple_check (PyObject *op)`
- `PyObject * py_tuple_new (size_t len)`
- `Py_ssize_t py_tuple_len (PyObject *op)`
- `int py_tuple_set (PyObject *tuple_p, size_t pos, PyObject *op)`
- `PyObject * py_tuple_get (PyObject *tuple_p, size_t pos)`
- `int py_list_check (PyObject *op)`
- `PyObject * py_list_new (size_t len)`
- `Py_ssize_t py_list_len (PyObject *op)`
- `int py_list_set (PyObject *list_p, size_t pos, PyObject *op)`
- `PyObject * py_list_get (PyObject *list_p, size_t pos)`
- `int py_set_check (PyObject *op)`
- `int py_frozenset_check (PyObject *op)`

- `template<template< typename ... > class ListLike, typename T , PyObject (*)(const T &) ConvertCppToPy, PyObject (*)(size_t) PyUnaryContainer_New, int (*)(PyObject *, size_t, PyObject *) PyUnaryContainer_Set>`
`PyObject * very_generic_cpp_std_list_like_to_py_unary (const ListLike< T > &list_like)`
- `template<template< typename ... > class ListLike, typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert, int (*)(PyObject *) PyUnaryContainer_Check, Py_ssize_t (*)(PyObject *) PyUnaryContainer_Size, PyObject (*)(PyObject *, size_t) PyUnaryContainer_Get>`
`int very_generic_py_unary_to_cpp_std_list_like (PyObject *op, ListLike< T > &list_like)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_list_like_to_py_tuple (const std::vector< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_list_like_to_py_tuple (const std::list< T > &container)`
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_tuple_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_tuple_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_list_like_to_py_list (const std::vector< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_list_like_to_py_list (const std::list< T > &container)`
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_list_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_list_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy, PyObject (*)(PyObject *) PyContainer_New>`
`PyObject * generic_cpp_std_unordered_set_to_py_set_or_frozenset (const std::unordered_set< T > &set)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_unordered_set_to_py_set (const std::unordered_set< T > &set)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_unordered_set_to_py_frozenset (const std::unordered_set< T > &set)`
- `template<typename T , int (*)(PyObject *) PyContainer_Check, int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_set_or_frozenset_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &set)`
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_set_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &set)`
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int generic_py_frozenset_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &set)`
- `template<template< typename ... > class Map, typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>`
`PyObject * generic_cpp_std_map_like_to_py_dict (const Map< K, V > &map)`
- `template<template< typename ... > class Map, typename K , typename V , int (*)(PyObject *) Check_K, int (*)(PyObject *) Check_V, K (*)(PyObject *) Convert_K, V (*)(PyObject *) Convert_V>`
`int generic_py_dict_to_cpp_std_map_like (PyObject *dict, Map< K, V > &map)`
- `template<template< typename ... > class ListLike, typename T , PyObject (*)(const T &) ConvertCppToPy, PyObject (*)(size_t) PyUnaryContainer_New, int (*)(PyObject *, size_t, PyObject *) PyUnaryContainer_Set>`
`PyObject * generic_cpp_std_list_like_to_py_list_like (const ListLike< T > &list_like)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_vector_to_py_tuple (const std::vector< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_list_to_py_tuple (const std::list< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_vector_to_py_list (const std::vector< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * generic_cpp_std_list_to_py_list (const std::list< T > &container)`
- `template<template< typename ... > class ListLike, typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert, int (*)(PyObject *) PyUnaryContainer_Check, Py_ssize_t (*)(PyObject *) PyUnaryContainer_Size, PyObject (*)(PyObject *, size_t) PyUnaryContainer_Get>`
`int generic_py_unary_to_cpp_std_list_like (PyObject *op, ListLike< T > &list_like)`

- `template<typename T, int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int generic_py_tuple_to_cpp_std_vector (PyObject *op, std::vector< T > &vec)`
- `template<typename T, int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int generic_py_list_to_cpp_std_vector (PyObject *op, std::vector< T > &vec)`
- `template<typename T, int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int generic_py_tuple_to_cpp_std_list (PyObject *op, std::list< T > &vec)`
- `template<typename T, int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int generic_py_list_to_cpp_std_list (PyObject *op, std::list< T > &vec)`
- `template<template< typename ... > class Map, typename K, typename V >`
`PyObject * cpp_std_map_like_to_py_dict (const std::unordered_map< K, V > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`PyObject * cpp_std_map_like_to_py_dict (const std::map< K, V > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`int py_dict_to_cpp_std_map_like (PyObject *op, std::unordered_map< K, V > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`int py_dict_to_cpp_std_map_like (PyObject *op, std::map< K, V > &map)`
- `PyObject * cpp_bool_to_py_bool (bool const &b)`
- `bool py_bool_to_cpp_bool (PyObject *op)`
- `int py_bool_check (PyObject *op)`
- `PyObject * cpp_long_to_py_long (const long &l)`
- `long py_long_to_cpp_long (PyObject *op)`
- `int py_long_check (PyObject *op)`
- `PyObject * cpp_double_to_py_float (const double &d)`
- `double py_float_to_cpp_double (PyObject *op)`
- `int py_float_check (PyObject *op)`
- `PyObject * cpp_complex_to_py_complex (const std::complex< double > &c)`
- `std::complex< double > py_complex_to_cpp_complex (PyObject *op)`
- `int py_complex_check (PyObject *op)`
- `PyObject * cpp_vector_char_to_py_bytes (const std::vector< char > &s)`
- `std::vector< char > py_bytes_to_cpp_vector_char (PyObject *op)`
- `int py_bytes_check (PyObject *op)`
- `PyObject * cpp_string_to_py_bytearray (const std::vector< char > &s)`
- `std::vector< char > py_bytearray_to_cpp_string (PyObject *op)`
- `int py_bytearray_check (PyObject *op)`
- `int py_unicode8_check (PyObject *op)`
- `PyObject * cpp_string_to_py_unicode8 (const std::string &s)`
- `std::string py_unicode8_to_cpp_string (PyObject *op)`
- `int py_unicode16_check (PyObject *op)`
- `PyObject * cpp_u16string_to_py_unicode16 (const std::u16string &s)`
- `std::u16string py_unicode16_to_cpp_u16string (PyObject *op)`
- `int py_unicode32_check (PyObject *op)`
- `PyObject * cpp_u32string_to_py_unicode32 (const std::u32string &s)`
- `std::u32string py_unicode32_to_cpp_u32string (PyObject *op)`
- `PyObject * cpp_vector_char_to_py_bytearray (const std::vector< char > &s)`
- `std::vector< char > py_bytearray_to_cpp_vector_char (PyObject *op)`
- `template<> PyObject * cpp_std_list_like_to_py_list< CppCustomObject > (const std::vector< CppCustomObject`
`> &container)`
- `template<> int py_list_to_cpp_std_list_like< CppCustomObject > (PyObject *op, std::vector< CppCustomObject`
`> &container)`
- `template<> PyObject * cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject > (const std::map< long,`
`CppCustomObject > &map)`
- `template<> int py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject > (PyObject *op, std::map< long,`
`CppCustomObject > &map)`

7.1.1 Detailed Description

Conversion functions for individual Python objects.

Functions to handle Python containers.

python_convert.cpp PythonC++

Created by Paul Ross on 22/11/2018. Copyright © 2018 Paul Ross. All rights reserved.

This contains the non-template hand maintained functions. Basically type conversion and checking functions.

7.1.2 Enumeration Type Documentation

7.1.2.1 ErrorReturnValue

```
enum Python_Cpp_Containers::ErrorReturnValue : int [strong]
```

Enumerator

SUCCESS	
FAIL_CONTAINER_WRONG_TYPE	
FAIL_CONTAINER_MEMBER_WRONG_TYPE	
FAIL_CONTAINER_KEY_WRONG_TYPE	
FAIL_CONTAINER_VALUE_WRONG_TYPE	

7.1.3 Function Documentation

7.1.3.1 cpp_bool_to_py_bool()

```
PyObject * Python_Cpp_Containers::cpp_bool_to_py_bool (
    bool const & b )
```

Converts a C++ `bool` to a Python `bool`. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.2 `cpp_complex_to_py_complex()`

```
PyObject * Python_Cpp_Containers::cpp_complex_to_py_complex (
    const std::complex< double > & c )
```

Converts a C++ `std::complex` to a Python `complex`. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.3 `cpp_double_to_py_float()`

```
PyObject * Python_Cpp_Containers::cpp_double_to_py_float (
    const double & d )
```

Converts a C++ `double` to a Python `float`. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.4 `cpp_long_to_py_long()`

```
PyObject * Python_Cpp_Containers::cpp_long_to_py_long (
    const long & l )
```

Converts a C++ `long` to a Python `long`. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.5 cpp_std_list_like_to_py_list() [1/2]

```
template<typename T >
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list (
    const std::list< T > & container )
```

Base declaration for converting C++ `std::list` to a Python `list`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>container</i>	C++ input as a <code>std::list<T></code> .
------------------	--

Returns

A Python `list` containing type corresponding to the C++ type `T`.

7.1.3.6 cpp_std_list_like_to_py_list() [2/2]

```
template<typename T >
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list (
    const std::vector< T > & container )
```

Base declaration for converting C++ `std::vector` to a Python `list`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>container</i>	C++ input as a <code>std::vector<T></code> .
------------------	--

Returns

A Python `list` containing type corresponding to the C++ type `T`.

7.1.3.7 `cpp_std_list_like_to_py_list< bool >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< bool > (
    const std::list< bool > & container )
```

Instantiation for converting C++ `std::list<bool>` to a Python list of bool.

Parameters

<i>container</i>	C++ input as a <code>std::list<bool></code> .
------------------	---

Returns

A Python list containing bool objects.

7.1.3.8 `cpp_std_list_like_to_py_list< bool >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< bool > (
    const std::vector< bool > & container )
```

Instantiation for converting C++ `std::vector<bool>` to a Python list of bool.

Parameters

<i>container</i>	C++ input as a <code>std::vector<bool></code> .
------------------	---

Returns

A Python list containing bool objects.

7.1.3.9 `cpp_std_list_like_to_py_list< CppCustomObject >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< CppCustomObject > (
    const std::vector< CppCustomObject > & container )
```

7.1.3.10 `cpp_std_list_like_to_py_list< double >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< double > (
    const std::list< double > & container )
```

Instantiation for converting C++ `std::list<double>` to a Python list of float.

Parameters

<i>container</i>	C++ input as a <code>std::list<double></code> .
------------------	---

Returns

A Python `list` containing `float` objects.

7.1.3.11 `cpp_std_list_like_to_py_list< double >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< double > (
    const std::vector< double > & container )
```

Instantiation for converting C++ `std::vector<double>` to a Python `list` of `float`.

Parameters

<i>container</i>	C++ input as a <code>std::vector<double></code> .
------------------	---

Returns

A Python `list` containing `float` objects.

7.1.3.12 `cpp_std_list_like_to_py_list< long >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (
    const std::list< long > & container )
```

Instantiation for converting C++ `std::list<long>` to a Python `list` of `int`.

Parameters

<i>container</i>	C++ input as a <code>std::list<long></code> .
------------------	---

Returns

A Python `list` containing `int` objects.

7.1.3.13 `cpp_std_list_like_to_py_list< long >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (
    const std::vector< long > & container )
```

Instantiation for converting C++ `std::vector<long>` to a Python list of int.

Parameters

<i>container</i>	C++ input as a <code>std::vector<long></code> .
------------------	---

Returns

A Python list containing int objects.

7.1.3.14 `cpp_std_list_like_to_py_list< std::complex< double > >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::complex< double > > (
    const std::list< std::complex< double > > & container )
```

Instantiation for converting C++ `std::list<std::complex<double>>` to a Python list of complex.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::complex<double>></code> .
------------------	---

Returns

A Python list containing complex objects.

7.1.3.15 `cpp_std_list_like_to_py_list< std::complex< double > >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::complex< double > > (
    const std::vector< std::complex< double > > & container )
```

Instantiation for converting C++ `std::vector<std::complex<double>>` to a Python list of complex.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::complex<double>></code> .
------------------	---

Returns

A Python `list` containing `complex` objects.

7.1.3.16 `cpp_std_list_like_to_py_list< std::string >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::string > (
    const std::list< std::string > & container )
```

Instantiation for converting C++ `std::list<std::string>` to a Python `list` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::string></code> .
------------------	--

Returns

A Python `list` containing `str` objects.

7.1.3.17 `cpp_std_list_like_to_py_list< std::string >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::string > (
    const std::vector< std::string > & container )
```

Instantiation for converting C++ `std::vector<std::string>` to a Python `list` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::string></code> .
------------------	--

Returns

A Python `list` containing `str` objects.

7.1.3.18 `cpp_std_list_like_to_py_list< std::u16string >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u16string > (
    const std::list< std::u16string > & container )
```

Instantiation for converting C++ `std::list<std::u16string>` to a Python `list` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::u16string></code> .
------------------	---

Returns

A Python `list` containing `str` objects.

7.1.3.19 `cpp_std_list_like_to_py_list< std::u16string >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u16string > (
    const std::vector< std::u16string > & container )
```

Instantiation for converting C++ `std::vector<std::u16string>` to a Python `list` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::u16string></code> .
------------------	---

Returns

A Python `list` containing `str` objects.

7.1.3.20 `cpp_std_list_like_to_py_list< std::u32string >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u32string > (
    const std::list< std::u32string > & container )
```

Instantiation for converting C++ `std::list<std::u32string>` to a Python `list` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::u32string></code> .
------------------	---

Returns

A Python `list` containing `str` objects.

7.1.3.21 `cpp_std_list_like_to_py_list< std::u32string >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u32string > (
    const std::vector< std::u32string > & container )
```

Instantiation for converting C++ `std::vector<std::u32string>` to a Python list of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::u32string></code> .
------------------	---

Returns

A Python list containing `str` objects.

7.1.3.22 `cpp_std_list_like_to_py_list< std::vector< char > >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::vector< char > > (
    const std::list< std::vector< char >> & container )
```

Instantiation for converting C++ `std::list<std::vector<char>>` to a Python list of bytes.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::vector<char>></code> .
------------------	--

Returns

A Python list containing bytes objects.

7.1.3.23 `cpp_std_list_like_to_py_list< std::vector< char > >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::vector< char > > (
    const std::vector< std::vector< char >> & container )
```

Instantiation for converting C++ `std::vector<std::vector<char>>` to a Python list of bytes.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::vector<char>></code> .
------------------	--

Returns

A Python `list` containing `bytes` objects.

7.1.3.24 `cpp_std_list_like_to_py_tuple()` [1/2]

```
template<typename T >
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple (
    const std::list< T > & container )
```

Base declaration for converting C++ `std::list` to a Python `tuple`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>container</i>	C++ input as a <code>std::list<T></code> .
------------------	--

Returns

A Python `tuple` containing type corresponding to the C++ type `T`.

7.1.3.25 `cpp_std_list_like_to_py_tuple()` [2/2]

```
template<typename T >
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple (
    const std::vector< T > & container )
```

Base declaration for converting C++ `std::vector` to a Python `tuple`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>container</i>	C++ input as a <code>std::vector<T></code> .
------------------	--

Returns

A Python `tuple` containing type corresponding to the C++ type `T`.

7.1.3.26 cpp_std_list_like_to_py_tuple< bool >() [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< bool > (
    const std::list< bool > & container )
```

Instantiation for converting C++ `std::list<bool>` to a Python tuple of bool.

Parameters

<i>container</i>	C++ input as a <code>std::list<bool></code> .
------------------	---

Returns

A Python tuple containing bool objects.

7.1.3.27 cpp_std_list_like_to_py_tuple< bool >() [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< bool > (
    const std::vector< bool > & container )
```

Instantiation for converting C++ `std::vector<bool>` to a Python tuple of bool.

Parameters

<i>container</i>	C++ input as a <code>std::vector<bool></code> .
------------------	---

Returns

A Python tuple containing bool objects.

7.1.3.28 cpp_std_list_like_to_py_tuple< double >() [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< double > (
    const std::list< double > & container )
```

Instantiation for converting C++ `std::list<double>` to a Python tuple of float.

Parameters

<i>container</i>	C++ input as a <code>std::list<double></code> .
------------------	---

Returns

A Python tuple containing float objects.

7.1.3.29 cpp_std_list_like_to_py_tuple< double >() [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< double > (
    const std::vector< double > & container )
```

Instantiation for converting C++ `std::vector<double>` to a Python tuple of float.

Parameters

<i>container</i>	C++ input as a <code>std::vector<double></code> .
------------------	---

Returns

A Python tuple containing float objects.

7.1.3.30 cpp_std_list_like_to_py_tuple< long >() [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (
    const std::list< long > & container )
```

Instantiation for converting C++ `std::list<long>` to a Python tuple of int.

Parameters

<i>container</i>	C++ input as a <code>std::list<long></code> .
------------------	---

Returns

A Python tuple containing int objects.

7.1.3.31 cpp_std_list_like_to_py_tuple< long >() [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (
    const std::vector< long > & container )
```

Instantiation for converting C++ `std::vector<long>` to a Python tuple of int.

Parameters

<i>container</i>	C++ input as a <code>std::vector<long></code> .
------------------	---

Returns

A Python tuple containing int objects.

7.1.3.32 `cpp_std_list_like_to_py_tuple< std::complex< double > >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::complex< double > > (
    const std::list< std::complex< double >> & container )
```

Instantiation for converting C++ `std::list<std::complex<double>>` to a Python tuple of complex.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::complex<double>></code> .
------------------	---

Returns

A Python tuple containing complex objects.

7.1.3.33 `cpp_std_list_like_to_py_tuple< std::complex< double > >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::complex< double > > (
    const std::vector< std::complex< double >> & container )
```

Instantiation for converting C++ `std::vector<std::complex<double>>` to a Python tuple of complex.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::complex<double>></code> .
------------------	---

Returns

A Python tuple containing complex objects.

7.1.3.34 `cpp_std_list_like_to_py_tuple< std::string >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::string > (
    const std::list< std::string > & container )
```

Instantiation for converting C++ `std::list<std::string>` to a Python tuple of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::string></code> .
------------------	--

Returns

A Python tuple containing `str` objects.

7.1.3.35 `cpp_std_list_like_to_py_tuple< std::string >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::string > (
    const std::vector< std::string > & container )
```

Instantiation for converting C++ `std::vector<std::string>` to a Python tuple of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::string></code> .
------------------	--

Returns

A Python tuple containing `str` objects.

7.1.3.36 `cpp_std_list_like_to_py_tuple< std::u16string >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u16string > (
    const std::list< std::u16string > & container )
```

Instantiation for converting C++ `std::list<std::u16string>` to a Python tuple of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::u16string></code> .
------------------	---

Returns

A Python tuple containing str objects.

7.1.3.37 cpp_std_list_like_to_py_tuple< std::u16string >() [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u16string > (
    const std::vector< std::u16string > & container )
```

Instantiation for converting C++ std::vector<std::u16string> to a Python tuple of str.

Parameters

<i>container</i>	C++ input as a std::vector<std::u16string>.
------------------	---

Returns

A Python tuple containing str objects.

7.1.3.38 cpp_std_list_like_to_py_tuple< std::u32string >() [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u32string > (
    const std::list< std::u32string > & container )
```

Instantiation for converting C++ std::list<std::u32string> to a Python tuple of str.

Parameters

<i>container</i>	C++ input as a std::list<std::u32string>.
------------------	---

Returns

A Python tuple containing str objects.

7.1.3.39 cpp_std_list_like_to_py_tuple< std::u32string >() [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u32string > (
    const std::vector< std::u32string > & container )
```

Instantiation for converting C++ std::vector<std::u32string> to a Python tuple of str.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::u32string></code> .
------------------	---

Returns

A Python tuple containing str objects.

7.1.3.40 `cpp_std_list_like_to_py_tuple< std::vector< char > >()` [1/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::vector< char > > (
    const std::list< std::vector< char >> & container )
```

Instantiation for converting C++ `std::list<std::vector<char>>` to a Python tuple of bytes.

Parameters

<i>container</i>	C++ input as a <code>std::list<std::vector<char>></code> .
------------------	--

Returns

A Python tuple containing bytes objects.

7.1.3.41 `cpp_std_list_like_to_py_tuple< std::vector< char > >()` [2/2]

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::vector< char > > (
    const std::vector< std::vector< char >> & container )
```

Instantiation for converting C++ `std::vector<std::vector<char>>` to a Python tuple of bytes.

Parameters

<i>container</i>	C++ input as a <code>std::vector<std::vector<char>></code> .
------------------	--

Returns

A Python tuple containing bytes objects.

7.1.3.42 cpp_std_map_like_to_py_dict() [1/3]

```
template<template< typename ... > class Map, typename K , typename V >
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict (
    const Map< K, V > & map )
```

Base declaration for converting a C++ `std::unordered_map<K, V>` to a Python dictionary.

Template Parameters

<i>K</i>	The C++ type for the key.
<i>V</i>	The C++ type for the value.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<K, V></code> .
------------	---

Returns

Python dictionary corresponding to `{K : V, ...}`.

7.1.3.43 cpp_std_map_like_to_py_dict() [2/3]

```
template<template< typename ... > class Map, typename K , typename V >
PyObject* Python_Cpp_Containers::cpp_std_map_like_to_py_dict (
    const std::map< K, V > & map )
```

7.1.3.44 cpp_std_map_like_to_py_dict() [3/3]

```
template<template< typename ... > class Map, typename K , typename V >
PyObject* Python_Cpp_Containers::cpp_std_map_like_to_py_dict (
    const std::unordered_map< K, V > & map )
```

7.1.3.45 cpp_std_map_like_to_py_dict< std::map, bool, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, bool > (
    const std::map< bool, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, bool></code> .
------------	---

Returns

A Python dictionary of {`bool` : `bool`, ...}.

7.1.3.46 `cpp_std_map_like_to_py_dict< std::map, bool, double >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, double > (
    const std::map< bool, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, double></code> .
------------	---

Returns

A Python dictionary of {`bool` : `float`, ...}.

7.1.3.47 `cpp_std_map_like_to_py_dict< std::map, bool, long >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, long > (
    const std::map< bool, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, long></code> .
------------	---

Returns

A Python dictionary of {`bool` : `int`, ...}.

7.1.3.48 cpp_std_map_like_to_py_dict< std::map, bool, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, std::complex<
double > > (
    const std::map< bool, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of {bool : complex, ...}.

7.1.3.49 cpp_std_map_like_to_py_dict< std::map, bool, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, std::string > (
    const std::map< bool, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::string></code> .
------------	--

Returns

A Python dictionary of {bool : str, ...}.

7.1.3.50 cpp_std_map_like_to_py_dict< std::map, bool, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, std::u16string
> (
    const std::map< bool, std::u16string > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::u16string></code> .
------------	---

Returns

A Python dictionary of {bool : str, ...}.

7.1.3.51 cpp_std_map_like_to_py_dict< std::map, bool, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, std::u32string
> (
    const std::map< bool, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::u32string></code> .
------------	---

Returns

A Python dictionary of {bool : str, ...}.

7.1.3.52 cpp_std_map_like_to_py_dict< std::map, bool, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, bool, std::vector<
char > > (
    const std::map< bool, std::vector< char >> & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {bool : bytes, ...}.

7.1.3.53 cpp_std_map_like_to_py_dict< std::map, double, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, bool > (
    const std::map< double, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, bool></code> .
------------	---

Returns

A Python dictionary of {float : bool, ...}.

7.1.3.54 cpp_std_map_like_to_py_dict< std::map, double, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, double > (
    const std::map< double, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, double></code> .
------------	---

Returns

A Python dictionary of {float : float, ...}.

7.1.3.55 cpp_std_map_like_to_py_dict< std::map, double, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, long > (
    const std::map< double, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, long></code> .
------------	---

Returns

A Python dictionary of {float : int, ...}.

7.1.3.56 cpp_std_map_like_to_py_dict< std::map, double, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::complex<
double > > (
    const std::map< double, std::complex< double >> & map )
```

Instantiation for converting a C++ std::unordered_map<double, std::complex<double>> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<double, std::complex<double>>.
------------	---

Returns

A Python dictionary of {float : complex, ...}.

7.1.3.57 cpp_std_map_like_to_py_dict< std::map, double, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::string >
(
    const std::map< double, std::string > & map )
```

Instantiation for converting a C++ std::unordered_map<double, std::string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<double, std::string>.
------------	--

Returns

A Python dictionary of {float : str, ...}.

7.1.3.58 cpp_std_map_like_to_py_dict< std::map, double, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::u16string
```

```
> (
    const std::map< double, std::u16string > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, std::u16string></code> .
------------	---

Returns

A Python dictionary of {float : str, ...}.

7.1.3.59 cpp_std_map_like_to_py_dict< std::map, double, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::u32string
> (
    const std::map< double, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, std::u32string></code> .
------------	---

Returns

A Python dictionary of {float : str, ...}.

7.1.3.60 cpp_std_map_like_to_py_dict< std::map, double, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::vector<
char > > (
    const std::map< double, std::vector< char >> & map )
```

Instantiation for converting a C++ `std::unordered_map<double, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {float : bytes, ...}.

7.1.3.61 cpp_std_map_like_to_py_dict< std::map, long, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, bool > (
    const std::map< long, bool > & map )
```

Instantiation for converting a C++ std::unordered_map<long, bool> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<long, bool>.
------------	---

Returns

A Python dictionary of {int : bool, ...}.

7.1.3.62 cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject
> (
    const std::map< long, CppCustomObject > & map )
```

7.1.3.63 cpp_std_map_like_to_py_dict< std::map, long, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, double > (
    const std::map< long, double > & map )
```

Instantiation for converting a C++ std::unordered_map<long, double> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<long, double>.
------------	---

Returns

A Python dictionary of {int : float, ...}.

7.1.3.64 cpp_std_map_like_to_py_dict< std::map, long, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, long > (
    const std::map< long, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, long></code> .
------------	---

Returns

A Python dictionary of {int : int, ...}.

7.1.3.65 cpp_std_map_like_to_py_dict< std::map, long, std::complex< double >>()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::complex<
double >> (
    const std::map< long, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of {int : complex, ...}.

7.1.3.66 cpp_std_map_like_to_py_dict< std::map, long, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::string > (
    const std::map< long, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::string></code> .
------------	--

Returns

A Python dictionary of {int : str, ...}.

7.1.3.67 cpp_std_map_like_to_py_dict< std::map, long, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::u16string
> (
    const std::map< long, std::u16string > & map )
```

Instantiation for converting a C++ std::unordered_map<long, std::u16string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<long, std::u16string>.
------------	---

Returns

A Python dictionary of {int : str, ...}.

7.1.3.68 cpp_std_map_like_to_py_dict< std::map, long, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::u32string
> (
    const std::map< long, std::u32string > & map )
```

Instantiation for converting a C++ std::unordered_map<long, std::u32string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<long, std::u32string>.
------------	---

Returns

A Python dictionary of {int : str, ...}.

7.1.3.69 cpp_std_map_like_to_py_dict< std::map, long, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::vector<
```



```
char > > (
    const std::map< long, std::vector< char >> & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {int : bytes, ...}.

7.1.3.70 cpp_std_map_like_to_py_dict< std::map, std::complex< double >, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::complex< double
>, bool > (
    const std::map< std::complex< double >, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::complex<double>, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::complex<double>, bool></code> .
------------	---

Returns

A Python dictionary of {complex : bool, ...}.

7.1.3.71 cpp_std_map_like_to_py_dict< std::map, std::complex< double >, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::complex< double
>, double > (
    const std::map< std::complex< double >, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::complex<double>, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::complex<double>, double></code> .
------------	---

Returns

A Python dictionary of {complex : float, ...}.

7.1.3.72 cpp_std_map_like_to_py_dict< std::map, std::complex< double >, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::complex< double
>, long > (
    const std::map< std::complex< double >, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::complex<double>, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::complex<double>, long></code> .
------------	---

Returns

A Python dictionary of {complex : int, ...}.

7.1.3.73 cpp_std_map_like_to_py_dict< std::map, std::string, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, bool > (
    const std::map< std::string, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, bool></code> .
------------	--

Returns

A Python dictionary of {str : bool, ...}.

7.1.3.74 cpp_std_map_like_to_py_dict< std::map, std::string, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, double >
```

```
(
    const std::map< std::string, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, double></code> .
------------	--

Returns

A Python dictionary of {str : float, ...}.

7.1.3.75 cpp_std_map_like_to_py_dict< std::map, std::string, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, long > (
    const std::map< std::string, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, long></code> .
------------	--

Returns

A Python dictionary of {str : int, ...}.

7.1.3.76 cpp_std_map_like_to_py_dict< std::map, std::string, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::
::complex< double > > (
    const std::map< std::string, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::complex<double>></code> .
------------	--

Returns

A Python dictionary of {str : complex, ...}.

7.1.3.77 cpp_std_map_like_to_py_dict< std::map, std::string, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::string > (
    const std::map< std::string, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.78 cpp_std_map_like_to_py_dict< std::map, std::string, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::u16string > (
    const std::map< std::string, std::u16string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::u16string></code> .
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.79 cpp_std_map_like_to_py_dict< std::map, std::string, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::u32string > (
    const std::map< std::string, std::u32string > & map )
```

```
::u32string > (
    const std::map< std::string, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::u32string></code> .
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.80 cpp_std_map_like_to_py_dict< std::map, std::string, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::
vector< char > > (
    const std::map< std::string, std::vector< char >> & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::vector<char>></code> .
------------	---

Returns

A Python dictionary of {str : bytes, ...}.

7.1.3.81 cpp_std_map_like_to_py_dict< std::map, std::u16string, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, bool
> (
    const std::map< std::u16string, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, bool></code> .
------------	---

Returns

A Python dictionary of {str : bool, ...}.

7.1.3.82 cpp_std_map_like_to_py_dict< std::map, std::u16string, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, double
> (
    const std::map< std::u16string, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, double></code> .
------------	---

Returns

A Python dictionary of {str : float, ...}.

7.1.3.83 cpp_std_map_like_to_py_dict< std::map, std::u16string, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, long
> (
    const std::map< std::u16string, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, long></code> .
------------	---

Returns

A Python dictionary of {str : int, ...}.

7.1.3.84 cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double >
```

```

::complex< double > > (
    const std::map< std::u16string, std::complex< double >> & map )

```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of {str : complex, ...}.

7.1.3.85 cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string >()

```

template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string > (
    const std::map< std::u16string, std::string > & map )

```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::string></code> .
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.86 cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string >()

```

template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string > (
    const std::map< std::u16string, std::u16string > & map )

```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::u16string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.87 cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string > (
    const std::map< std::u16string, std::u32string > & map )
```

Instantiation for converting a C++ std::unordered_map<std::u16string, std::u32string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u16string, std::u32string>.
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.88 cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > > (
    const std::map< std::u16string, std::vector< char > > & map )
```

Instantiation for converting a C++ std::unordered_map<std::u16string, std::vector<char>> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u16string, std::vector<char>>.
------------	--

Returns

A Python dictionary of {str : bytes, ...}.

7.1.3.89 cpp_std_map_like_to_py_dict< std::map, std::u32string, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, bool
```



```
> (
    const std::map< std::u32string, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, bool></code> .
------------	---

Returns

A Python dictionary of {str : bool, ...}.

7.1.3.90 cpp_std_map_like_to_py_dict< std::map, std::u32string, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, double
> (
    const std::map< std::u32string, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, double></code> .
------------	---

Returns

A Python dictionary of {str : float, ...}.

7.1.3.91 cpp_std_map_like_to_py_dict< std::map, std::u32string, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, long
> (
    const std::map< std::u32string, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, long></code> .
------------	---

Returns

A Python dictionary of {str : int, ...}.

7.1.3.92 cpp_std_map_like_to_py_dict< std::map, std::u32string, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::
::complex< double > > (
    const std::map< std::u32string, std::complex< double >> & map )
```

Instantiation for converting a C++ std::unordered_map<std::u32string, std::complex<double>> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u32string, std::complex<double>>.
------------	---

Returns

A Python dictionary of {str : complex, ...}.

7.1.3.93 cpp_std_map_like_to_py_dict< std::map, std::u32string, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::
::string > (
    const std::map< std::u32string, std::string > & map )
```

Instantiation for converting a C++ std::unordered_map<std::u32string, std::string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u32string, std::string>.
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.94 cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::
```

```

::ul6string > (
    const std::map< std::u32string, std::ul6string > & map )

```

Instantiation for converting a C++ `std::unordered_map<std::u32string, std::ul6string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, std::ul6string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.95 cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u32string >()

```

template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u32string > (
    const std::map< std::u32string, std::u32string > & map )

```

Instantiation for converting a C++ `std::unordered_map<std::u32string, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, std::u32string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.96 cpp_std_map_like_to_py_dict< std::map, std::u32string, std::vector< char > >()

```

template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::vector< char > > (
    const std::map< std::u32string, std::vector< char > > & map )

```

Instantiation for converting a C++ `std::unordered_map<std::u32string, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {str : bytes, ...}.

7.1.3.97 cpp_std_map_like_to_py_dict< std::map, std::vector< char >, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::vector< char >,
bool > (
    const std::map< std::vector< char >, bool > & map )
```

Instantiation for converting a C++ std::unordered_map<std::vector<char>, bool> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::vector<char>, bool>.
------------	--

Returns

A Python dictionary of {bytes : bool, ...}.

7.1.3.98 cpp_std_map_like_to_py_dict< std::map, std::vector< char >, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::vector< char >,
double > (
    const std::map< std::vector< char >, double > & map )
```

Instantiation for converting a C++ std::unordered_map<std::vector<char>, double> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::vector<char>, double>.
------------	--

Returns

A Python dictionary of {bytes : float, ...}.

7.1.3.99 cpp_std_map_like_to_py_dict< std::map, std::vector< char >, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::vector< char >,
```

```
long > (
    const std::map< std::vector< char >, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::vector<char>, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::vector<char>, long></code> .
------------	--

Returns

A Python dictionary of {bytes : int, ...}.

7.1.3.100 cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool
> (
    const std::unordered_map< bool, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, bool></code> .
------------	---

Returns

A Python dictionary of {bool : bool, ...}.

7.1.3.101 cpp_std_map_like_to_py_dict< std::unordered_map, bool, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, double
> (
    const std::unordered_map< bool, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, double></code> .
------------	---

Returns

A Python dictionary of {bool : float, ...}.

7.1.3.102 cpp_std_map_like_to_py_dict< std::unordered_map, bool, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, long
> (
    const std::unordered_map< bool, long > & map )
```

Instantiation for converting a C++ std::unordered_map<bool, long> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<bool, long>.
------------	---

Returns

A Python dictionary of {bool : int, ...}.

7.1.3.103 cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::
::complex< double > > (
    const std::unordered_map< bool, std::complex< double >> & map )
```

Instantiation for converting a C++ std::unordered_map<bool, std::complex<double>> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<bool, std::complex<double>>.
------------	---

Returns

A Python dictionary of {bool : complex, ...}.

7.1.3.104 cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::
::string > (
    const std::unordered_map< bool, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::string></code> .
------------	--

Returns

A Python dictionary of `{bool : str, ...}`.

7.1.3.105 `cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string > (
    const std::unordered_map< bool, std::u16string > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::u16string></code> .
------------	---

Returns

A Python dictionary of `{bool : str, ...}`.

7.1.3.106 `cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string > (
    const std::unordered_map< bool, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<bool, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<bool, std::u32string></code> .
------------	---

Returns

A Python dictionary of {bool : str, ...}.

7.1.3.107 cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > > (
    const std::unordered_map< bool, std::vector< char >> & map )
```

Instantiation for converting a C++ std::unordered_map<bool, std::vector<char>> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<bool, std::vector<char>>.
------------	--

Returns

A Python dictionary of {bool : bytes, ...}.

7.1.3.108 cpp_std_map_like_to_py_dict< std::unordered_map, double, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
bool > (
    const std::unordered_map< double, bool > & map )
```

Instantiation for converting a C++ std::unordered_map<double, bool> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<double, bool>.
------------	---

Returns

A Python dictionary of {float : bool, ...}.

7.1.3.109 cpp_std_map_like_to_py_dict< std::unordered_map, double, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
```



```
double > (
    const std::unordered_map< double, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, double></code> .
------------	---

Returns

A Python dictionary of {float : float, ...}.

7.1.3.110 cpp_std_map_like_to_py_dict< std::unordered_map, double, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
long > (
    const std::unordered_map< double, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, long></code> .
------------	---

Returns

A Python dictionary of {float : int, ...}.

7.1.3.111 cpp_std_map_like_to_py_dict< std::unordered_map, double, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
std::complex< double > > (
    const std::unordered_map< double, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<double, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of {float : complex, ...}.

7.1.3.112 cpp_std_map_like_to_py_dict< std::unordered_map, double, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
std::string > (
    const std::unordered_map< double, std::string > & map )
```

Instantiation for converting a C++ std::unordered_map<double, std::string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<double, std::string>.
------------	--

Returns

A Python dictionary of {float : str, ...}.

7.1.3.113 cpp_std_map_like_to_py_dict< std::unordered_map, double, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
std::u16string > (
    const std::unordered_map< double, std::u16string > & map )
```

Instantiation for converting a C++ std::unordered_map<double, std::u16string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<double, std::u16string>.
------------	---

Returns

A Python dictionary of {float : str, ...}.

7.1.3.114 cpp_std_map_like_to_py_dict< std::unordered_map, double, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
```

```
std::u32string > (
    const std::unordered_map< double, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<double, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, std::u32string></code> .
------------	---

Returns

A Python dictionary of {float : str, ...}.

7.1.3.115 cpp_std_map_like_to_py_dict< std::unordered_map, double, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double,
std::vector< char > > (
    const std::unordered_map< double, std::vector< char >> & map )
```

Instantiation for converting a C++ `std::unordered_map<double, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<double, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {float : bytes, ...}.

7.1.3.116 cpp_std_map_like_to_py_dict< std::unordered_map, long, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, bool
> (
    const std::unordered_map< long, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, bool></code> .
------------	---

Returns

A Python dictionary of {int : bool, ...}.

7.1.3.117 cpp_std_map_like_to_py_dict< std::unordered_map, long, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, double
> (
    const std::unordered_map< long, double > & map )
```

Instantiation for converting a C++ std::unordered_map<long, double> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<long, double>.
------------	---

Returns

A Python dictionary of {int : float, ...}.

7.1.3.118 cpp_std_map_like_to_py_dict< std::unordered_map, long, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, long
> (
    const std::unordered_map< long, long > & map )
```

Instantiation for converting a C++ std::unordered_map<long, long> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<long, long>.
------------	---

Returns

A Python dictionary of {int : int, ...}.

7.1.3.119 cpp_std_map_like_to_py_dict< std::unordered_map, long, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::
::complex< double > > (
    const std::unordered_map< long, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of `{int : complex, ...}`.

7.1.3.120 `cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string > (
    const std::unordered_map< long, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::string></code> .
------------	--

Returns

A Python dictionary of `{int : str, ...}`.

7.1.3.121 `cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string > (
    const std::unordered_map< long, std::u16string > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::u16string></code> .
------------	---

Returns

A Python dictionary of {int : str, ...}.

7.1.3.122 cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u32string > (
    const std::unordered_map< long, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::u32string></code> .
------------	---

Returns

A Python dictionary of {int : str, ...}.

7.1.3.123 cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > > (
    const std::unordered_map< long, std::vector< char > > & map )
```

Instantiation for converting a C++ `std::unordered_map<long, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<long, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {int : bytes, ...}.

7.1.3.124 cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::complex<
```

```
double >, bool > (
    const std::unordered_map< std::complex< double >, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::complex<double>, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::complex<double>, bool></code> .
------------	---

Returns

A Python dictionary of {complex : bool, ...}.

7.1.3.125 `cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, double >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::complex<
double >, double > (
    const std::unordered_map< std::complex< double >, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::complex<double>, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::complex<double>, double></code> .
------------	---

Returns

A Python dictionary of {complex : float, ...}.

7.1.3.126 `cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, long >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::complex<
double >, long > (
    const std::unordered_map< std::complex< double >, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::complex<double>, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::complex<double>, long></code> .
------------	---

Returns

A Python dictionary of {complex : int, ...}.

7.1.3.127 cpp_std_map_like_to_py_dict< std::unordered_map, std::string, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
bool > (
    const std::unordered_map< std::string, bool > & map )
```

Instantiation for converting a C++ std::unordered_map<std::string, bool> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::string, bool>.
------------	--

Returns

A Python dictionary of {str : bool, ...}.

7.1.3.128 cpp_std_map_like_to_py_dict< std::unordered_map, std::string, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
double > (
    const std::unordered_map< std::string, double > & map )
```

Instantiation for converting a C++ std::unordered_map<std::string, double> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::string, double>.
------------	--

Returns

A Python dictionary of {str : float, ...}.

7.1.3.129 cpp_std_map_like_to_py_dict< std::unordered_map, std::string, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
```



```
long > (
    const std::unordered_map< std::string, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, long></code> .
------------	--

Returns

A Python dictionary of {str : int, ...}.

7.1.3.130 `cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::complex< double > >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
std::complex< double > > (
    const std::unordered_map< std::string, std::complex< double > > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::complex<double>></code> .
------------	--

Returns

A Python dictionary of {str : complex, ...}.

7.1.3.131 `cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
std::string > (
    const std::unordered_map< std::string, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.132 cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
std::u16string > (
    const std::unordered_map< std::string, std::u16string > & map )
```

Instantiation for converting a C++ std::unordered_map<std::string, std::u16string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::string, std::u16string>.
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.133 cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
std::u32string > (
    const std::unordered_map< std::string, std::u32string > & map )
```

Instantiation for converting a C++ std::unordered_map<std::string, std::u32string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::string, std::u32string>.
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.134 cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::string,
```

```
std::vector< char > > (
    const std::unordered_map< std::string, std::vector< char >> & map )
```

Instantiation for converting a C++ `std::unordered_map<std::string, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::string, std::vector<char>></code> .
------------	---

Returns

A Python dictionary of {str : bytes, ...}.

7.1.3.135 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
bool > (
    const std::unordered_map< std::u16string, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, bool></code> .
------------	---

Returns

A Python dictionary of {str : bool, ...}.

7.1.3.136 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
double > (
    const std::unordered_map< std::u16string, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, double></code> .
------------	---

Returns

A Python dictionary of {str : float, ...}.

7.1.3.137 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
long > (
    const std::unordered_map< std::u16string, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, long></code> .
------------	---

Returns

A Python dictionary of {str : int, ...}.

7.1.3.138 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::complex< double > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
std::complex< double > > (
    const std::unordered_map< std::u16string, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of {str : complex, ...}.

7.1.3.139 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
```

```
std::string > (
    const std::unordered_map< std::u16string, std::string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::string></code> .
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.140 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
std::u16string > (
    const std::unordered_map< std::u16string, std::u16string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::u16string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::u16string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.141 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
std::u32string > (
    const std::unordered_map< std::u16string, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u16string, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u16string, std::u32string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.142 cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string,
std::vector< char > > (
    const std::unordered_map< std::u16string, std::vector< char >> & map )
```

Instantiation for converting a C++ std::unordered_map<std::u16string, std::vector<char>> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u16string, std::vector<char>>.
------------	--

Returns

A Python dictionary of {str : bytes, ...}.

7.1.3.143 cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
bool > (
    const std::unordered_map< std::u32string, bool > & map )
```

Instantiation for converting a C++ std::unordered_map<std::u32string, bool> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u32string, bool>.
------------	---

Returns

A Python dictionary of {str : bool, ...}.

7.1.3.144 cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
```

```
double > (
    const std::unordered_map< std::u32string, double > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, double>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, double></code> .
------------	---

Returns

A Python dictionary of {str : float, ...}.

7.1.3.145 `cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, long >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
long > (
    const std::unordered_map< std::u32string, long > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, long>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, long></code> .
------------	---

Returns

A Python dictionary of {str : int, ...}.

7.1.3.146 `cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::complex< double > >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
std::complex< double > > (
    const std::unordered_map< std::u32string, std::complex< double >> & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, std::complex<double>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, std::complex<double>></code> .
------------	---

Returns

A Python dictionary of {str : complex, ...}.

7.1.3.147 cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
std::string > (
    const std::unordered_map< std::u32string, std::string > & map )
```

Instantiation for converting a C++ std::unordered_map<std::u32string, std::string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u32string, std::string>.
------------	--

Returns

A Python dictionary of {str : str, ...}.

7.1.3.148 cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u16string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
std::u16string > (
    const std::unordered_map< std::u32string, std::u16string > & map )
```

Instantiation for converting a C++ std::unordered_map<std::u32string, std::u16string> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::u32string, std::u16string>.
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.149 cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u32string >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
```



```
std::u32string > (
    const std::unordered_map< std::u32string, std::u32string > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, std::u32string>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, std::u32string></code> .
------------	---

Returns

A Python dictionary of {str : str, ...}.

7.1.3.150 cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::vector< char > >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string,
std::vector< char > > (
    const std::unordered_map< std::u32string, std::vector< char > > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::u32string, std::vector<char>>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::u32string, std::vector<char>></code> .
------------	--

Returns

A Python dictionary of {str : bytes, ...}.

7.1.3.151 cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, bool >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::vector<
char >, bool > (
    const std::unordered_map< std::vector< char >, bool > & map )
```

Instantiation for converting a C++ `std::unordered_map<std::vector<char>, bool>` to a Python dictionary.

Parameters

<i>map</i>	Input C++ <code>std::unordered_map<std::vector<char>, bool></code> .
------------	--

Returns

A Python dictionary of {bytes : bool, ...}.

7.1.3.152 cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, double >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::vector<
char >, double > (
    const std::unordered_map< std::vector< char >, double > & map )
```

Instantiation for converting a C++ std::unordered_map<std::vector<char>, double> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::vector<char>, double>.
------------	--

Returns

A Python dictionary of {bytes : float, ...}.

7.1.3.153 cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, long >()

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, std::vector<
char >, long > (
    const std::unordered_map< std::vector< char >, long > & map )
```

Instantiation for converting a C++ std::unordered_map<std::vector<char>, long> to a Python dictionary.

Parameters

<i>map</i>	Input C++ std::unordered_map<std::vector<char>, long>.
------------	--

Returns

A Python dictionary of {bytes : int, ...}.

7.1.3.154 cpp_std_unordered_set_to_py_frozenset()

```
template<typename T >
PyObject* Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset (
    const std::unordered_set< T > & container )
```

Base declaration for converting C++ `std::unordered_set` to a Python `frozenset`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<T></code> .
------------------	---

Returns

A Python `frozenset` containing type corresponding to the C++ type `T`.

7.1.3.155 `cpp_std_unordered_set_to_py_frozenset< bool >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< bool > (
    const std::unordered_set< bool > & container )
```

Instantiation for converting C++ `std::unordered_set<bool>` to a Python `frozenset` of `bool`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<bool></code> .
------------------	--

Returns

A Python `frozenset` containing `bool` objects.

7.1.3.156 `cpp_std_unordered_set_to_py_frozenset< double >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< double > (
    const std::unordered_set< double > & container )
```

Instantiation for converting C++ `std::unordered_set<double>` to a Python `frozenset` of `float`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<double></code> .
------------------	--

Returns

A Python `frozenset` containing `float` objects.

7.1.3.157 `cpp_std_unordered_set_to_py_frozenset< long >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< long > (
    const std::unordered_set< long > & container )
```

Instantiation for converting C++ `std::unordered_set<long>` to a Python `frozenset` of `int`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<long></code> .
------------------	--

Returns

A Python `frozenset` containing `int` objects.

7.1.3.158 `cpp_std_unordered_set_to_py_frozenset< std::complex< double > >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::complex< double
> > (
    const std::unordered_set< std::complex< double >> & container )
```

Instantiation for converting C++ `std::unordered_set<std::complex<double>>` to a Python `frozenset` of `complex`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::complex<double>></code> .
------------------	--

Returns

A Python `frozenset` containing `complex` objects.

7.1.3.159 `cpp_std_unordered_set_to_py_frozenset< std::string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::string > (
    const std::unordered_set< std::string > & container )
```

Instantiation for converting C++ `std::unordered_set<std::string>` to a Python `frozenset` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::string></code> .
------------------	---

Returns

A Python `frozenset` containing `str` objects.

7.1.3.160 `cpp_std_unordered_set_to_py_frozenset< std::u16string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::u16string > (
    const std::unordered_set< std::u16string > & container )
```

Instantiation for converting C++ `std::unordered_set<std::u16string>` to a Python `frozenset` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::u16string></code> .
------------------	--

Returns

A Python `frozenset` containing `str` objects.

7.1.3.161 `cpp_std_unordered_set_to_py_frozenset< std::u32string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::u32string > (
    const std::unordered_set< std::u32string > & container )
```

Instantiation for converting C++ `std::unordered_set<std::u32string>` to a Python `frozenset` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::u32string></code> .
------------------	--

Returns

A Python `frozenset` containing `str` objects.

7.1.3.162 `cpp_std_unordered_set_to_py_frozenset< std::vector< char >>()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::vector< char > >
> (
    const std::unordered_set< std::vector< char >> & container )
```

Instantiation for converting C++ `std::unordered_set<std::vector<char>>` to a Python `frozenset` of bytes.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::vector<char>></code> .
------------------	---

Returns

A Python `frozenset` containing bytes objects.

7.1.3.163 `cpp_std_unordered_set_to_py_set()`

```
template<typename T >
PyObject* Python_Cpp_Containers::cpp_std_unordered_set_to_py_set (
    const std::unordered_set< T > & container )
```

Base declaration for converting C++ `std::unordered_set` to a Python `set`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<T></code> .
------------------	---

Returns

A Python `set` containing type corresponding to the C++ type `T`.

7.1.3.164 `cpp_std_unordered_set_to_py_set< bool >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< bool > (
    const std::unordered_set< bool > & container )
```

Instantiation for converting C++ `std::unordered_set<bool>` to a Python `set` of `bool`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<bool></code> .
------------------	--

Returns

A Python set containing `bool` objects.

7.1.3.165 `cpp_std_unordered_set_to_py_set< double >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< double > (
    const std::unordered_set< double > & container )
```

Instantiation for converting C++ `std::unordered_set<double>` to a Python set of float.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<double></code> .
------------------	--

Returns

A Python set containing `float` objects.

7.1.3.166 `cpp_std_unordered_set_to_py_set< long >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< long > (
    const std::unordered_set< long > & container )
```

Instantiation for converting C++ `std::unordered_set<long>` to a Python set of int.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<long></code> .
------------------	--

Returns

A Python set containing `int` objects.

7.1.3.167 `cpp_std_unordered_set_to_py_set< std::complex< double > >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::complex< double > > (
    const std::unordered_set< std::complex< double >> & container )
```

Instantiation for converting C++ `std::unordered_set<std::complex<double>>` to a Python set of complex.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::complex<double>></code> .
------------------	--

Returns

A Python set containing complex objects.

7.1.3.168 `cpp_std_unordered_set_to_py_set< std::string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::string > (
    const std::unordered_set< std::string > & container )
```

Instantiation for converting C++ `std::unordered_set<std::string>` to a Python set of str.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::string></code> .
------------------	---

Returns

A Python set containing str objects.

7.1.3.169 `cpp_std_unordered_set_to_py_set< std::u16string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::u16string > (
    const std::unordered_set< std::u16string > & container )
```

Instantiation for converting C++ `std::unordered_set<std::u16string>` to a Python set of str.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::u16string></code> .
------------------	--

Returns

A Python `set` containing `str` objects.

7.1.3.170 `cpp_std_unordered_set_to_py_set< std::u32string >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::u32string > (
    const std::unordered_set< std::u32string > & container )
```

Instantiation for converting C++ `std::unordered_set<std::u32string>` to a Python `set` of `str`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::u32string></code> .
------------------	--

Returns

A Python `set` containing `str` objects.

7.1.3.171 `cpp_std_unordered_set_to_py_set< std::vector< char > >()`

```
template<>
PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::vector< char > > (
    const std::unordered_set< std::vector< char > > & container )
```

Instantiation for converting C++ `std::unordered_set<std::vector<char>>` to a Python `set` of `bytes`.

Parameters

<i>container</i>	C++ input as a <code>std::unordered_set<std::vector<char>></code> .
------------------	---

Returns

A Python `set` containing `bytes` objects.

7.1.3.172 `cpp_string_to_py_bytearray()`

```
PyObject* Python_Cpp_Containers::cpp_string_to_py_bytearray (
    const std::vector< char > & s )
```

7.1.3.173 `cpp_string_to_py_unicode8()`

```
PyObject * Python_Cpp_Containers::cpp_string_to_py_unicode8 (
    const std::string & s )
```

Create a Python 16 bit unicode object from a C++ `std::string`.

Converts a C++ `std::string` to a Python `str`. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.174 `cpp_u16string_to_py_unicode16()`

```
PyObject * Python_Cpp_Containers::cpp_u16string_to_py_unicode16 (
    const std::u16string & s )
```

Create a Python 16 bit unicode object from a C++ `std::u16string`.

NOTE: We can't use `PyUnicode_FromKindAndData(PyUnicode_2BYTE_KIND, (Py_UCS2 *)s.c_str(), s.size())` since that function may produce a `PyUnicode_1BYTE_KIND` if the character values permit. Instead we use `PyUnicode_New(s.size(), maxchar = 65535)` to compel a 2 byte word size and then copy each character individually.

See: https://docs.python.org/3/c-api/unicode.html#c.PyUnicode_New

Parameters

<i>s</i>	The C++ string.
----------	-----------------

Returns

The Python string.

Converts a C++ `std::string` to a Python `str` with `PyUnicode_2BYTE_KIND` entries. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.175 cpp_u32string_to_py_unicode32()

```
PyObject * Python_Cpp_Containers::cpp_u32string_to_py_unicode32 (
    const std::u32string & s )
```

Create a Python 32 bit unicode object from a C++ std::u32string.

NOTE: We can't use `PyUnicode_FromKindAndData(PyUnicode_4BYTE_KIND, (Py_UCS4 *)s.c_str(), s.size())` since that function may produce a `PyUnicode_1BYTE_KIND` or `PyUnicode_2BYTE_KIND` if the character values permit. Instead we use `PyUnicode_New(s.size(), maxchar = 1114111)` to compel a 4 byte word size and then copy each character individually.

See: https://docs.python.org/3/c-api/unicode.html#c.PyUnicode_New

Parameters

<i>s</i>	The C++ string.
----------	-----------------

Returns

The Python string.

Converts a C++ `std::string` to a Python `str` with `PyUnicode_2BYTE_KIND` entries. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.176 cpp_vector_char_to_py_bytearray()

```
PyObject* Python_Cpp_Containers::cpp_vector_char_to_py_bytearray (
    const std::vector< char > & s )
```

Converts a C++ `std::vector<char>` to a Python `bytearray`. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.177 `cpp_vector_char_to_py_bytes()`

```
PyObject * Python_Cpp_Containers::cpp_vector_char_to_py_bytes (
    const std::vector< char > & s )
```

Converts a C++ `std::vector<char>` to a Python bytes. This always succeeds.

Parameters

<i>b</i>	Value to convert.
----------	-------------------

Returns

Value equivalent in Python.

7.1.3.178 `generic_cpp_std_list_like_to_py_list()` [1/2]

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_like_to_py_list (
    const std::list< T > & container )
```

Partial specialisation of the template to convert from a C++ `std::list<T>` to a Python list.

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>ConvertCppToPy</i>	Pointer to a conversion function to convert a C++ type <code>T</code> to an equivalent Python type.

Parameters

<i>vec</i>	The C++ <code>std::list<T></code> as input.
------------	---

Returns

A new Python list with the contents of the input.

7.1.3.179 generic_cpp_std_list_like_to_py_list() [2/2]

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_like_to_py_list (
    const std::vector< T > & container )
```

Partial specialisation of the template to convert from a C++ `std::vector<T>` to a Python list.

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>ConvertCppToPy</i>	Pointer to a conversion function to convert a C++ type T to an equivalent Python type.

Parameters

<i>vec</i>	The C++ <code>std::vector<T></code> as input.
------------	---

Returns

A new Python list with the contents of the input.

7.1.3.180 generic_cpp_std_list_like_to_py_list_like()

```
template<template< typename ... > class ListLike, typename T , PyObject *(*)(const T &)
ConvertCppToPy, PyObject *(*)(size_t) PyUnaryContainer_New, int(*)(PyObject *, size_t, PyObject
*) PyUnaryContainer_Set>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_like_to_py_list_like (
    const ListLike< T > & list_like )
```

7.1.3.181 generic_cpp_std_list_like_to_py_tuple() [1/2]

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_like_to_py_tuple (
    const std::list< T > & container )
```

Partial specialisation of the template to convert from a C++ `std::list<T>` to a Python tuple.

Example complete specialisation for C++ long using this:

```
template<>
PyObject *
cpp_std_vector_to_py_tuple<long>(const std::list<long> &container) {
    return generic_cpp_std_vector_to_py_tuple<long, &cpp_long_to_py_long>(container);
}
```

Then this can be used thus:

```
std::list<long> cpp_list;
// Populate cpp_list...
//
// Convert to a Python tuple of int.
PyObject *op = Python_Cpp_Containers::cpp_std_vector_to_py_tuple(cpp_list);
// If op == NULL then a Python error will be set.
```

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>ConvertCppToPy</i>	Pointer to a conversion function to convert a C++ type T to an equivalent Python type.

Parameters

<i>container</i>	The C++ <code>std::list<T></code> as input.
------------------	---

Returns

A new Python tuple with the contents of the input or NULL on failure in which case a `PyErr...` will be set.

7.1.3.182 generic_cpp_std_list_like_to_py_tuple() [2/2]

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_like_to_py_tuple (
    const std::vector< T > & container )
```

Partial specialisation of the template to convert from a C++ `std::vector<T>` to a Python tuple.

Example complete specialisation for C++ `long` using this:

```
template <>
PyObject *
cpp_std_vector_to_py_tuple<long>(const std::vector<long> &container) {
    return generic_cpp_std_vector_to_py_tuple<long, &cpp_long_to_py_long>(container);
}
```

Then this can be used thus:

```
std::vector<long> cpp_vector;
// Populate cpp_vector...
//
// Convert to a Python tuple of int.
PyObject *op = Python_Cpp_Containers::cpp_std_vector_to_py_tuple(cpp_vector);
// If op == NULL then a Python error will be set.
```

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>ConvertCppToPy</i>	Pointer to a conversion function to convert a C++ type T to an equivalent Python type.

Parameters

<i>container</i>	The C++ <code>std::vector<T></code> as input.
------------------	---

Returns

A new Python tuple with the contents of the input or NULL on failure in which case a `PyErr...` will be set.

7.1.3.183 generic_cpp_std_list_to_py_list()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_to_py_list (
    const std::list< T > & container )
```

7.1.3.184 generic_cpp_std_list_to_py_tuple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_list_to_py_tuple (
    const std::list< T > & container )
```

7.1.3.185 generic_cpp_std_map_like_to_py_dict()

```
template<template< typename ... > class Map, typename K , typename V , PyObject *(*)(const K
&) Convert_K, PyObject *(*)(const V &) Convert_V>
PyObject * Python_Cpp_Containers::generic_cpp_std_map_like_to_py_dict (
    const Map< K, V > & map )
```

This is a hand written generic function to convert a C++ `unordered_map` to a Python dict. The template is instantiated with C++ type(s) and a conversion function(s) to create Python object(s) from those types.

Example:

```
template <>
PyObject *
cpp_std_map_like_to_py_dict<std::unordered_map, double, double>(const std::unordered_map<double, double>
&map) {
    return generic_cpp_std_map_like_to_py_dict<
        double, double,
        &cpp_double_to_py_float, &cpp_double_to_py_float
    >(map);
}
```

Template Parameters

<i>Map</i>	The container, either <code>std::unordered_map</code> or <code>std::map</code> .
<i>K</i>	The C++ type of the key.
<i>V</i>	The C++ type of the value.
<i>Convert_K</i>	A function to convert a C++ type K to a PyObject*.
<i>Convert_V</i>	A function to convert a C++ type V to a PyObject*.

Parameters

<i>map</i>	The C++ <code>std::unordered_map</code> that is to be converted to a Python dict.
------------	---

Returns

The Python dictionary. NULL on failure.

7.1.3.186 generic_cpp_std_unordered_set_to_py_frozenset()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_unordered_set_to_py_frozenset (
    const std::unordered_set< T > & set )
```

Specific instantiation to convert a C++ `std::unordered_set<T>` to a Python `frozenset`.

Template Parameters

<i>T</i>	The C++ type of the objects in the <code>std::unordered_set</code> .
<i>ConvertCppToPy</i>	Function to convert the C++ <code><T></code> to a <code>PyObject*</code> .

Parameters

<i>set</i>	The C++ <code>std::unordered_set</code> as input data.
------------	--

Returns

The `PyObject*` `frozenset` containing the values of the C++ `std::unordered_set`.

7.1.3.187 generic_cpp_std_unordered_set_to_py_set()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_unordered_set_to_py_set (
    const std::unordered_set< T > & set )
```

Specific instantiation to convert a C++ `std::unordered_set<T>` to a Python `set`.

Template Parameters

<i>T</i>	The C++ type of the objects in the <code>std::unordered_set</code> .
<i>ConvertCppToPy</i>	Function to convert the C++ <code><T></code> to a <code>PyObject*</code> .

Parameters

<i>set</i>	The C++ <code>std::unordered_set</code> as input data.
------------	--

Returns

The PyObject* set containing the values of the C++ std::unordered_set.

7.1.3.188 generic_cpp_std_unordered_set_to_py_set_or_frozenset()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy, PyObject *(*)(PyObject *) Py↵
Container_New>
PyObject* Python_Cpp_Containers::generic_cpp_std_unordered_set_to_py_set_or_frozenset (
    const std::unordered_set< T > & set )
```

This is a hand written generic function to convert a C++ std::unordered_set<T> to a Python set or frozenset.

Template Parameters

<i>T</i>	The C++ type of the objects in the std::unordered_set.
<i>ConvertCppToPy</i>	Function to convert the C++ <T> to a PyObject*.
<i>PyContainer_New</i>	Function to create a new Python container.

Parameters

<i>set</i>	The C++ std::unordered_set as input data.
------------	---

Returns

The PyObject* set or frozenset containing the values of the C++ std::unordered_set.

7.1.3.189 generic_cpp_std_vector_to_py_list()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_vector_to_py_list (
    const std::vector< T > & container )
```

7.1.3.190 generic_cpp_std_vector_to_py_tuple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
PyObject* Python_Cpp_Containers::generic_cpp_std_vector_to_py_tuple (
    const std::vector< T > & container )
```

7.1.3.191 generic_py_dict_to_cpp_std_map_like()

```
template<template< typename ... > class Map, typename K , typename V , int(*) (PyObject *)
Check_K, int(*) (PyObject *) Check_V, K(*) (PyObject *) Convert_K, V(*) (PyObject *) Convert_V>
int Python_Cpp_Containers::generic_py_dict_to_cpp_std_map_like (
    PyObject * dict,
    Map< K, V > & map )
```

This is a hand written generic function to convert a Python dict to a C++ `std::unordered_map` or `std::map`.

Template Parameters

<i>Map</i>	The container, either <code>std::unordered_map</code> or <code>std::map</code> .
<i>K</i>	The C++ type of the key.
<i>V</i>	The C++ type of the value.
<i>Check_K</i>	A function to check that the type of a <code>PyObject*</code> is a C++ <code>K</code> . Returns 0 on success, non-zero on failure.
<i>Check_V</i>	A function to check that the type of a <code>PyObject*</code> is a C++ <code>V</code> . Returns 0 on success, non-zero on failure.
<i>Convert_K</i>	A function to convert a <code>PyObject*</code> to a C++ type <code>K</code> .
<i>Convert_V</i>	A function to convert a <code>PyObject*</code> to a C++ type <code>V</code> .

Parameters

<i>dict</i>	The Python dictionary as input.
<i>map</i>	The C++ <code>std::unordered_map</code> to populate. This will be empty on failure.

Returns

0 on success. Non-zero on failure.

7.1.3.192 generic_py_frozenset_to_cpp_std_unordered_set()

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_frozenset_to_cpp_std_unordered_set (
    PyObject * op,
    std::unordered_set< T > & set )
```

Specific instantiation to convert a Python `frozenset` to a C++ `std::unordered_set<T>`.

Template Parameters

<i>T</i>	The C++ type of the objects in the <code>std::unordered_set</code> .
<i>PyObject_Check</i>	A pointer to a function that checks that a member of the Python container is the correct type.
<i>PyObject_Convert</i>	A pointer to a function that converts a Python object to the C++ object of type <code>T</code> .

Parameters

<i>op</i>	The Python container,
<i>set</i>	The C++ result.

Returns

Zero on success, non-zero on failure.

7.1.3.193 generic_py_list_to_cpp_std_list()

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_list_to_cpp_std_list (
    PyObject * op,
    std::list< T > & vec )
```

7.1.3.194 generic_py_list_to_cpp_std_list_like() [1/2]

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_list_to_cpp_std_list_like (
    PyObject * op,
    std::list< T > & container )
```

Partial specialisation of the template to convert from a Python list to a C++ `std::list<T>`.

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>PyObject_Check</i>	Pointer to a function that checks the types of the objects in the list can be converted to a C++ type T.
<i>PyObject_Convert</i>	Pointer to a function that converts a Python object to a C++ type T.

Parameters

<i>op</i>	The Python list as input.
<i>vec</i>	The C++ <code>std::list<T></code> as output. This will be empty on failure.

Returns

Zero on success, non-zero on failure.

7.1.3.195 generic_py_list_to_cpp_std_list_like() [2/2]

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_list_to_cpp_std_list_like (
    PyObject * op,
    std::vector< T > & container )
```

Partial specialisation of the template to convert from a Python `list` to a C++ `std::vector<T>`.

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>PyObject_Check</i>	Pointer to a function that checks the types of the objects in the list can be converted to a C++ type T.
<i>PyObject_Convert</i>	Pointer to a function that converts a Python object to a C++ type T.

Parameters

<i>op</i>	The Python list as input.
<i>vec</i>	The C++ <code>std::vector<T></code> as output. This will be empty on failure.

Returns

Zero on success, non-zero on failure.

7.1.3.196 generic_py_list_to_cpp_std_vector()

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_list_to_cpp_std_vector (
    PyObject * op,
    std::vector< T > & vec )
```

7.1.3.197 generic_py_set_or_frozenset_to_cpp_std_unordered_set()

```
template<typename T , int(*) (PyObject *) PyContainer_Check, int(*) (PyObject *) PyObject_Check,
T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_set_or_frozenset_to_cpp_std_unordered_set (
    PyObject * op,
    std::unordered_set< T > & set )
```

This is a hand written generic function to convert a Python `set` or `frozenset` to a C++ `std::unordered_set<T>`.

Template Parameters

<i>T</i>	The C++ type of the objects in the <code>std::unordered_set</code> .
<i>PyContainer_Check</i>	A pointer to a function that checks that the Python container is the correct type.
<i>PyObject_Check</i>	A pointer to a function that checks that a member of the Python container is the correct type.
<i>PyObject_Convert</i>	A pointer to a function that converts a Python object to the C++ object of type T.

Parameters

<i>op</i>	The Python container,
<i>set</i>	The C++ result.

Returns

Zero on success, non-zero on failure.

7.1.3.198 generic_py_set_to_cpp_std_unordered_set()

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_set_to_cpp_std_unordered_set (
    PyObject * op,
    std::unordered_set< T > & set )
```

Specific instantiation to convert a Python set to a C++ `std::unordered_set<T>`.

Template Parameters

<i>T</i>	The C++ type of the objects in the <code>std::unordered_set</code> .
<i>PyObject_Check</i>	A pointer to a function that checks that a member of the Python container is the correct type.
<i>PyObject_Convert</i>	A pointer to a function that converts a Python object to the C++ object of type <code>T</code> .

Parameters

<i>op</i>	The Python container,
<i>set</i>	The C++ result.

Returns

Zero on success, non-zero on failure.

7.1.3.199 generic_py_tuple_to_cpp_std_list()

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_list (
    PyObject * op,
    std::list< T > & vec )
```

7.1.3.200 generic_py_tuple_to_cpp_std_list_like() [1/2]

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_list_like (
    PyObject * op,
    std::list< T > & container )
```

Partial specialisation of the template to convert from a Python `tuple` to a C++ `std::list<T>`.

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>PyObject_Check</i>	Pointer to a function that checks the types of the objects in the tuple can be converted to a C++ type T.
<i>PyObject_Convert</i>	Pointer to a function that converts a Python object to a C++ type T.

Parameters

<i>op</i>	The Python <code>tuple</code> as input.
<i>vec</i>	The C++ <code>std::list<T></code> as output. This will be empty on failure.

Returns

Zero on success, non-zero on failure.

7.1.3.201 generic_py_tuple_to_cpp_std_list_like() [2/2]

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_list_like (
    PyObject * op,
    std::vector< T > & container )
```

Partial specialisation of the template to convert from a Python `tuple` to a C++ `std::vector<T>`.

Template Parameters

<i>T</i>	C++ type of the objects in the container.
<i>PyObject_Check</i>	Pointer to a function that checks the types of the objects in the tuple can be converted to a C++ type T.
<i>PyObject_Convert</i>	Pointer to a function that converts a Python object to a C++ type T.

Parameters

<i>op</i>	The Python <code>tuple</code> as input.
<i>vec</i>	The C++ <code>std::vector<T></code> as output. This will be empty on failure.

Returns

Zero on success, non-zero on failure.

7.1.3.202 generic_py_tuple_to_cpp_std_vector()

```
template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>
int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_vector (
    PyObject * op,
    std::vector< T > & vec )
```

7.1.3.203 generic_py_unary_to_cpp_std_list_like()

```
template<template< typename ... > class ListLike, typename T , int(*) (PyObject *) PyObject_↵
Check, T(*) (PyObject *) PyObject_Convert, int(*) (PyObject *) PyUnaryContainer_Check, Py_ssize_↵
_t(*) (PyObject *) PyUnaryContainer_Size, PyObject *(*)(PyObject *, size_t) PyUnaryContainer_↵
Get>
int Python_Cpp_Containers::generic_py_unary_to_cpp_std_list_like (
    PyObject * op,
    ListLike< T > & list_like )
```

7.1.3.204 py_bool_check()

```
int Python_Cpp_Containers::py_bool_check (
    PyObject * op )
```

Return non-zero if the given value is a Python `bool` type.

Parameters

<i>op</i>	The Python object to check to be a bool type.
-----------	---

Returns

Zero if not a Python `bool`, non-zero if a Python `bool`.

7.1.3.205 py_bool_to_cpp_bool()

```
bool Python_Cpp_Containers::py_bool_to_cpp_bool (
    PyObject * op )
```

Converts a Python `bool` to a C++ `bool`. This asserts that the given value is a Python `bool`. If asserts are enabled then this asserts that the argument is a Python `bool` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

true or false.

7.1.3.206 py_bytearray_check()

```
int Python_Cpp_Containers::py_bytearray_check (
    PyObject * op )
```

Return non-zero if the given value is a Python bytearray type.

Parameters

<i>op</i>	The Python object to check to be a bytearray type.
-----------	--

Returns

Zero if not a Python bytearray, non-zero if a Python bytearray.

7.1.3.207 py_bytearray_to_cpp_string()

```
std::vector<char> Python_Cpp_Containers::py_bytearray_to_cpp_string (
    PyObject * op )
```

7.1.3.208 py_bytearray_to_cpp_vector_char()

```
std::vector<char> Python_Cpp_Containers::py_bytearray_to_cpp_vector_char (
    PyObject * op )
```

Converts a Python bytearray to a C++ `std::vector<char>`. This asserts that the given value is a Python bytearray. If asserts are enabled then this asserts that the argument is a Python bytearray objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `std::vector<char>`.

7.1.3.209 py_bytes_check()

```
int Python_Cpp_Containers::py_bytes_check (
    PyObject * op )
```

Return non-zero if the given value is a Python `bytes` type.

Parameters

<i>op</i>	The Python object to check to be a <code>bytes</code> type.
-----------	---

Returns

Zero if not a Python `bytes`, non-zero if a Python `bytes`.

7.1.3.210 py_bytes_to_cpp_vector_char()

```
std::vector< char > Python_Cpp_Containers::py_bytes_to_cpp_vector_char (
    PyObject * op )
```

Converts a Python `bytes` to a C++ `std::vector<char>`. This asserts that the given value is a Python `bytes`. If asserts are enabled then this asserts that the argument is a Python `bytes` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `std::vector<char>`.

7.1.3.211 py_complex_check()

```
int Python_Cpp_Containers::py_complex_check (
    PyObject * op )
```

Return non-zero if the given value is a Python `complex` type.

Parameters

<i>op</i>	The Python object to check to be a <code>complex</code> type.
-----------	---

Returns

Zero if not a Python `complex`, non-zero if a Python `complex`.

7.1.3.212 py_complex_to_cpp_complex()

```
std::complex< double > Python_Cpp_Containers::py_complex_to_cpp_complex (
    PyObject * op )
```

Converts a Python `complex` to a C++ `std::complex`. This asserts that the given value is a Python `complex`. If asserts are enabled then this asserts that the argument is a Python `complex` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `std::complex`.

7.1.3.213 py_dict_to_cpp_std_map_like() [1/3]

```
template<template< typename ... > class Map, typename K , typename V >
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like (
    PyObject * op,
    Map< K, V > & map )
```

Base declaration for converting a Python dictionary to a C++ `std::unordered_map<K, V>`.

Template Parameters

<i>K</i>	The C++ type for the key.
<i>V</i>	The C++ type for the value.

Parameters

<i>op</i>	The Python dictionary as the input.
<i>map</i>	C++ <code>std::unordered_map<K, V></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.214 py_dict_to_cpp_std_map_like() [2/3]

```
template<template< typename ... > class Map, typename K , typename V >
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like (
    PyObject * op,
    std::map< K, V > & map )
```

7.1.3.215 py_dict_to_cpp_std_map_like() [3/3]

```
template<template< typename ... > class Map, typename K , typename V >
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like (
    PyObject * op,
    std::unordered_map< K, V > & map )
```

7.1.3.216 py_dict_to_cpp_std_map_like< std::map, bool, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, bool > (
    PyObject * op,
    std::map< bool, bool > & map )
```

Instantiation for converting a Python dictionary {bool : bool, ...} to a C++ std::unordered_map<bool, bool>.

Parameters

<i>op</i>	A Python dictionary of {bool : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.217 py_dict_to_cpp_std_map_like< std::map, bool, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, double > (
```

```
PyObject * op,
std::map< bool, double > & map )
```

Instantiation for converting a Python dictionary {bool : float, ...} to a C++ std::unordered_map<bool, double>.

Parameters

<i>op</i>	A Python dictionary of {bool : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.218 py_dict_to_cpp_std_map_like< std::map, bool, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, long > (
    PyObject * op,
    std::map< bool, long > & map )
```

Instantiation for converting a Python dictionary {bool : int, ...} to a C++ std::unordered_map<bool, long>.

Parameters

<i>op</i>	A Python dictionary of {bool : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.219 py_dict_to_cpp_std_map_like< std::map, bool, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, std::complex< double
> > (
    PyObject * op,
    std::map< bool, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {bool : complex, ...} to a C++ std::unordered_map<bool, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {bool : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.220 py_dict_to_cpp_std_map_like< std::map, bool, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, std::string > (
    PyObject * op,
    std::map< bool, std::string > & map )
```

Instantiation for converting a Python dictionary {bool : str, ...} to a C++ std::unordered_map<bool, std::string>.

Parameters

<i>op</i>	A Python dictionary of {bool : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.221 py_dict_to_cpp_std_map_like< std::map, bool, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, std::u16string > (
    PyObject * op,
    std::map< bool, std::u16string > & map )
```

Instantiation for converting a Python dictionary {bool : str, ...} to a C++ std::unordered_map<bool, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {bool : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.222 py_dict_to_cpp_std_map_like< std::map, bool, std::u32string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, std::u32string > (
    PyObject * op,
    std::map< bool, std::u32string > & map )
```

Instantiation for converting a Python dictionary {bool : str, ...} to a C++ std::unordered_map<bool, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {bool : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.223 py_dict_to_cpp_std_map_like< std::map, bool, std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, bool, std::vector< char > > (
    PyObject * op,
    std::map< bool, std::vector< char > > & map )
```

Instantiation for converting a Python dictionary {bool : bytes, ...} to a C++ std::unordered_map<bool, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {bool : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.224 py_dict_to_cpp_std_map_like< std::map, double, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, bool > (
    PyObject * op,
    std::map< double, bool > & map )
```

Instantiation for converting a Python dictionary {float : bool, ...} to a C++ std::unordered_map<double, bool>.

Parameters

<i>op</i>	A Python dictionary of {float : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.225 py_dict_to_cpp_std_map_like< std::map, double, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, double > (
    PyObject * op,
    std::map< double, double > & map )
```

Instantiation for converting a Python dictionary {float : float, ...} to a C++ std::unordered_map<double, double>.

Parameters

<i>op</i>	A Python dictionary of {float : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.226 py_dict_to_cpp_std_map_like< std::map, double, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, long > (
    PyObject * op,
    std::map< double, long > & map )
```

Instantiation for converting a Python dictionary {float : int, ...} to a C++ std::unordered_map<double, long>.

Parameters

<i>op</i>	A Python dictionary of {float : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.227 py_dict_to_cpp_std_map_like< std::map, double, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::complex< double
> > (
    PyObject * op,
    std::map< double, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {float : complex, ...} to a C++ std::unordered_map<double, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {float : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.228 py_dict_to_cpp_std_map_like< std::map, double, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::string > (
    PyObject * op,
    std::map< double, std::string > & map )
```

Instantiation for converting a Python dictionary {float : str, ...} to a C++ std::unordered_map<double, std::string>.

Parameters

<i>op</i>	A Python dictionary of {float : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.229 py_dict_to_cpp_std_map_like< std::map, double, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::u16string > (
    PyObject * op,
    std::map< double, std::u16string > & map )
```

Instantiation for converting a Python dictionary {float : str, ...} to a C++ std::unordered_map<double, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {float : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.230 py_dict_to_cpp_std_map_like< std::map, double, std::u32string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::u32string > (
    PyObject * op,
    std::map< double, std::u32string > & map )
```

Instantiation for converting a Python dictionary {float : str, ...} to a C++ std::unordered_map<double, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {float : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.231 `py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::vector< char >
> (
    PyObject * op,
    std::map< double, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {float : bytes, ...} to a C++ `std::unordered_map<double, std::vector<char>>`.

Parameters

<i>op</i>	A Python dictionary of {float : bytes, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<double, std::vector<char>></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.232 `py_dict_to_cpp_std_map_like< std::map, long, bool >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, bool > (
    PyObject * op,
    std::map< long, bool > & map )
```

Instantiation for converting a Python dictionary {int : bool, ...} to a C++ `std::unordered_map<long, bool>`.

Parameters

<i>op</i>	A Python dictionary of {int : bool, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<long, bool></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.233 `py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject > (
    PyObject * op,
    std::map< long, CppCustomObject > & map )
```

7.1.3.234 py_dict_to_cpp_std_map_like< std::map, long, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, double > (
    PyObject * op,
    std::map< long, double > & map )
```

Instantiation for converting a Python dictionary {int : float, ...} to a C++ std::unordered_map<long, double>.

Parameters

<i>op</i>	A Python dictionary of {int : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.235 py_dict_to_cpp_std_map_like< std::map, long, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, long > (
    PyObject * op,
    std::map< long, long > & map )
```

Instantiation for converting a Python dictionary {int : int, ...} to a C++ std::unordered_map<long, long>.

Parameters

<i>op</i>	A Python dictionary of {int : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.236 py_dict_to_cpp_std_map_like< std::map, long, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::complex< double
> > (
    PyObject * op,
    std::map< long, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {int : complex, ...} to a C++ std::unordered_map<long, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {int : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.237 py_dict_to_cpp_std_map_like< std::map, long, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::string > (
    PyObject * op,
    std::map< long, std::string > & map )
```

Instantiation for converting a Python dictionary {int : str, ...} to a C++ std::unordered_map<long, std::string>.

Parameters

<i>op</i>	A Python dictionary of {int : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.238 py_dict_to_cpp_std_map_like< std::map, long, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::u16string > (
    PyObject * op,
    std::map< long, std::u16string > & map )
```

Instantiation for converting a Python dictionary {int : str, ...} to a C++ std::unordered_map<long, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {int : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.239 py_dict_to_cpp_std_map_like< std::map, long, std::u32string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::u32string > (
    PyObject * op,
    std::map< long, std::u32string > & map )
```

Instantiation for converting a Python dictionary {int : str, ...} to a C++ std::unordered_map<long, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {int : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.240 py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > >
(
    PyObject * op,
    std::map< long, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {int : bytes, ...} to a C++ std::unordered_map<long, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {int : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.241 `py_dict_to_cpp_std_map_like< std::map, std::complex< double >, bool >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::complex< double >,
bool > (
    PyObject * op,
    std::map< std::complex< double >, bool > & map )
```

Instantiation for converting a Python dictionary {complex : bool, ...} to a C++ `std::unordered_map<std::complex<double>, bool>`.

Parameters

<i>op</i>	A Python dictionary of {complex : bool, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::complex<double>, bool></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.242 `py_dict_to_cpp_std_map_like< std::map, std::complex< double >, double >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::complex< double >,
double > (
    PyObject * op,
    std::map< std::complex< double >, double > & map )
```

Instantiation for converting a Python dictionary {complex : float, ...} to a C++ `std::unordered_map<std::complex<double>, double>`.

Parameters

<i>op</i>	A Python dictionary of {complex : float, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::complex<double>, double></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.243 `py_dict_to_cpp_std_map_like< std::map, std::complex< double >, long >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::complex< double >,
long > (
```

```
PyObject * op,
std::map< std::complex< double >, long > & map )
```

Instantiation for converting a Python dictionary {complex : int, ...} to a C++ std::unordered_map<std::complex<double>, long>.

Parameters

<i>op</i>	A Python dictionary of {complex : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::complex<double>, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.244 py_dict_to_cpp_std_map_like< std::map, std::string, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, bool > (
    PyObject * op,
    std::map< std::string, bool > & map )
```

Instantiation for converting a Python dictionary {str : bool, ...} to a C++ std::unordered_map<std::string, bool>.

Parameters

<i>op</i>	A Python dictionary of {str : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.245 py_dict_to_cpp_std_map_like< std::map, std::string, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, double > (
    PyObject * op,
    std::map< std::string, double > & map )
```

Instantiation for converting a Python dictionary {str : float, ...} to a C++ std::unordered_map<std::string, double>.

Parameters

<i>op</i>	A Python dictionary of {str : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.246 py_dict_to_cpp_std_map_like< std::map, std::string, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, long > (
    PyObject * op,
    std::map< std::string, long > & map )
```

Instantiation for converting a Python dictionary {str : int, ...} to a C++ std::unordered_map<std::string, long>.

Parameters

<i>op</i>	A Python dictionary of {str : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.247 py_dict_to_cpp_std_map_like< std::map, std::string, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::complex<
double > > (
    PyObject * op,
    std::map< std::string, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {str : complex, ...} to a C++ std::unordered_map<std::string, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {str : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.248 py_dict_to_cpp_std_map_like< std::map, std::string, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::string > (
    PyObject * op,
    std::map< std::string, std::string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::string, std::string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.249 py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string
> (
    PyObject * op,
    std::map< std::string, std::u16string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::string, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.250 `py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string
> (
    PyObject * op,
    std::map< std::string, std::u32string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::string, std::u32string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::string, std::u32string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.251 `py_dict_to_cpp_std_map_like< std::map, std::string, std::vector< char > >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::vector<
char > > (
    PyObject * op,
    std::map< std::string, std::vector< char > > & map )
```

Instantiation for converting a Python dictionary {str : bytes, ...} to a C++ `std::unordered_map<std::string, std::vector<char>>`.

Parameters

<i>op</i>	A Python dictionary of {str : bytes, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::string, std::vector<char>></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.252 `py_dict_to_cpp_std_map_like< std::map, std::u16string, bool >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, bool > (
```

```
PyObject * op,  
std::map< std::u16string, bool > & map )
```

Instantiation for converting a Python dictionary {str : bool, ...} to a C++ std::unordered_map<std::u16string, bool>.

Parameters

<i>op</i>	A Python dictionary of {str : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.253 py_dict_to_cpp_std_map_like< std::map, std::u16string, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, double > (
    PyObject * op,
    std::map< std::u16string, double > & map )
```

Instantiation for converting a Python dictionary {str : float, ...} to a C++ std::unordered_map<std::u16string, double>.

Parameters

<i>op</i>	A Python dictionary of {str : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.254 py_dict_to_cpp_std_map_like< std::map, std::u16string, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, long > (
    PyObject * op,
    std::map< std::u16string, long > & map )
```

Instantiation for converting a Python dictionary {str : int, ...} to a C++ std::unordered_map<std::u16string, long>.

Parameters

<i>op</i>	A Python dictionary of {str : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.255 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex<
double > > (
    PyObject * op,
    std::map< std::u16string, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {str : complex, ...} to a C++ std::unordered_map<std::u16string, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {str : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.256 py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string
> (
    PyObject * op,
    std::map< std::u16string, std::string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::u16string, std::string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.257 `py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string
> (
    PyObject * op,
    std::map< std::u16string, std::u16string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::u16string, std::u16string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u16string, std::u16string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.258 `py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string
> (
    PyObject * op,
    std::map< std::u16string, std::u32string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::u16string, std::u32string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u16string, std::u32string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.259 `py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector< char > >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector<
char > > > (
```

```
PyObject * op,
std::map< std::u16string, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {str : bytes, ...} to a C++ std::unordered_map<std::u16string, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {str : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.260 py_dict_to_cpp_std_map_like< std::map, std::u32string, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, bool > (
    PyObject * op,
    std::map< std::u32string, bool > & map )
```

Instantiation for converting a Python dictionary {str : bool, ...} to a C++ std::unordered_map<std::u32string, bool>.

Parameters

<i>op</i>	A Python dictionary of {str : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.261 py_dict_to_cpp_std_map_like< std::map, std::u32string, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, double > (
    PyObject * op,
    std::map< std::u32string, double > & map )
```

Instantiation for converting a Python dictionary {str : float, ...} to a C++ std::unordered_map<std::u32string, double>.

Parameters

<i>op</i>	A Python dictionary of {str : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.262 py_dict_to_cpp_std_map_like< std::map, std::u32string, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, long > (
    PyObject * op,
    std::map< std::u32string, long > & map )
```

Instantiation for converting a Python dictionary {str : int, ...} to a C++ std::unordered_map<std::u32string, long>.

Parameters

<i>op</i>	A Python dictionary of {str : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.263 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex<
double > > (
    PyObject * op,
    std::map< std::u32string, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {str : complex, ...} to a C++ std::unordered_map<std::u32string, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {str : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.264 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string
> (
    PyObject * op,
    std::map< std::u32string, std::string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::u32string, std::string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.265 py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string
> (
    PyObject * op,
    std::map< std::u32string, std::u16string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::u32string, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.266 `py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string
> (
    PyObject * op,
    std::map< std::u32string, std::u32string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::u32string, std::u32string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u32string, std::u32string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.267 `py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector< char > >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector<
char > > (
    PyObject * op,
    std::map< std::u32string, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {str : bytes, ...} to a C++ `std::unordered_map<std::u32string, std::vector<char>>`.

Parameters

<i>op</i>	A Python dictionary of {str : bytes, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u32string, std::vector<char>></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.268 `py_dict_to_cpp_std_map_like< std::map, std::vector< char >, bool >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::vector< char >, bool >
(
```

```
PyObject * op,
std::map< std::vector< char >, bool > & map )
```

Instantiation for converting a Python dictionary {bytes : bool, ...} to a C++ `std::unordered_map<std::vector<char>, bool>`.

Parameters

<i>op</i>	A Python dictionary of {bytes : bool, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::vector<char>, bool></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.269 py_dict_to_cpp_std_map_like< std::map, std::vector< char >, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::vector< char >, double
> (
    PyObject * op,
    std::map< std::vector< char >, double > & map )
```

Instantiation for converting a Python dictionary {bytes : float, ...} to a C++ `std::unordered_map<std::vector<char>, double>`.

Parameters

<i>op</i>	A Python dictionary of {bytes : float, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::vector<char>, double></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.270 py_dict_to_cpp_std_map_like< std::map, std::vector< char >, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::vector< char >, long >
(
    PyObject * op,
    std::map< std::vector< char >, long > & map )
```

Instantiation for converting a Python dictionary {bytes : int, ...} to a C++ `std::unordered_map<std::vector<char>, long>`.

Parameters

<i>op</i>	A Python dictionary of {bytes : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::vector<char>, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.271 py_dict_to_cpp_std_map_like< std::unordered_map, bool, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, bool > (
    PyObject * op,
    std::unordered_map< bool, bool > & map )
```

Instantiation for converting a Python dictionary {bool : bool, ...} to a C++ std::unordered_map<bool, bool>.

Parameters

<i>op</i>	A Python dictionary of {bool : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.272 py_dict_to_cpp_std_map_like< std::unordered_map, bool, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, double > (
    PyObject * op,
    std::unordered_map< bool, double > & map )
```

Instantiation for converting a Python dictionary {bool : float, ...} to a C++ std::unordered_map<bool, double>.

Parameters

<i>op</i>	A Python dictionary of {bool : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.273 py_dict_to_cpp_std_map_like< std::unordered_map, bool, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, long > (
    PyObject * op,
    std::unordered_map< bool, long > & map )
```

Instantiation for converting a Python dictionary {bool : int, ...} to a C++ std::unordered_map<bool, long>.

Parameters

<i>op</i>	A Python dictionary of {bool : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.274 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::complex<
double > > (
    PyObject * op,
    std::unordered_map< bool, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {bool : complex, ...} to a C++ std::unordered_map<bool, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {bool : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.275 `py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::string
> (
    PyObject * op,
    std::unordered_map< bool, std::string > & map )
```

Instantiation for converting a Python dictionary {bool : str, ...} to a C++ std::unordered_map<bool, std::string>.

Parameters

<i>op</i>	A Python dictionary of {bool : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.276 `py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u16string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u16string
> (
    PyObject * op,
    std::unordered_map< bool, std::u16string > & map )
```

Instantiation for converting a Python dictionary {bool : str, ...} to a C++ std::unordered_map<bool, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {bool : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.277 `py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u32string
> (
```

```
PyObject * op,
std::unordered_map< bool, std::u32string > & map )
```

Instantiation for converting a Python dictionary {bool : str, ...} to a C++ std::unordered_map<bool, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {bool : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.278 py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::vector<
char > > (
    PyObject * op,
    std::unordered_map< bool, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {bool : bytes, ...} to a C++ std::unordered_map<bool, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {bool : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<bool, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.279 py_dict_to_cpp_std_map_like< std::unordered_map, double, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, bool > (
    PyObject * op,
    std::unordered_map< double, bool > & map )
```

Instantiation for converting a Python dictionary {float : bool, ...} to a C++ std::unordered_map<double, bool>.

Parameters

<i>op</i>	A Python dictionary of {float : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.280 py_dict_to_cpp_std_map_like< std::unordered_map, double, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, double > (
    PyObject * op,
    std::unordered_map< double, double > & map )
```

Instantiation for converting a Python dictionary {float : float, ...} to a C++ std::unordered_map<double, double>.

Parameters

<i>op</i>	A Python dictionary of {float : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.281 py_dict_to_cpp_std_map_like< std::unordered_map, double, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, long > (
    PyObject * op,
    std::unordered_map< double, long > & map )
```

Instantiation for converting a Python dictionary {float : int, ...} to a C++ std::unordered_map<double, long>.

Parameters

<i>op</i>	A Python dictionary of {float : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.282 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::complex< double > > (
    PyObject * op,
    std::unordered_map< double, std::complex< double > > & map )
```

Instantiation for converting a Python dictionary {float : complex, ...} to a C++ std::unordered_map<double, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {float : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.283 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::string > (
    PyObject * op,
    std::unordered_map< double, std::string > & map )
```

Instantiation for converting a Python dictionary {float : str, ...} to a C++ std::unordered_map<double, std::string>.

Parameters

<i>op</i>	A Python dictionary of {float : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.284 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u16string > (
    PyObject * op,
    std::unordered_map< double, std::u16string > & map )
```

Instantiation for converting a Python dictionary {float : str, ...} to a C++ std::unordered_map<double, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {float : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.285 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u32string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u32string > (
    PyObject * op,
    std::unordered_map< double, std::u32string > & map )
```

Instantiation for converting a Python dictionary {float : str, ...} to a C++ std::unordered_map<double, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {float : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.286 py_dict_to_cpp_std_map_like< std::unordered_map, double, std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::vector< char > > (
    PyObject * op,
    std::unordered_map< double, std::vector< char > > & map )
```

```
PyObject * op,
std::unordered_map< double, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {float : bytes, ...} to a C++ std::unordered_map<double, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {float : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<double, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.287 py_dict_to_cpp_std_map_like< std::unordered_map, long, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, bool > (
    PyObject * op,
    std::unordered_map< long, bool > & map )
```

Instantiation for converting a Python dictionary {int : bool, ...} to a C++ std::unordered_map<long, bool>.

Parameters

<i>op</i>	A Python dictionary of {int : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.288 py_dict_to_cpp_std_map_like< std::unordered_map, long, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, double > (
    PyObject * op,
    std::unordered_map< long, double > & map )
```

Instantiation for converting a Python dictionary {int : float, ...} to a C++ std::unordered_map<long, double>.

Parameters

<i>op</i>	A Python dictionary of {int : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.289 py_dict_to_cpp_std_map_like< std::unordered_map, long, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, long > (
    PyObject * op,
    std::unordered_map< long, long > & map )
```

Instantiation for converting a Python dictionary {int : int, ...} to a C++ std::unordered_map<long, long>.

Parameters

<i>op</i>	A Python dictionary of {int : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.290 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex<
double > > (
    PyObject * op,
    std::unordered_map< long, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {int : complex, ...} to a C++ std::unordered_map<long, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {int : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.291 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::string
> (
    PyObject * op,
    std::unordered_map< long, std::string > & map )
```

Instantiation for converting a Python dictionary {int : str, ...} to a C++ std::unordered_map<long, std::string>.

Parameters

<i>op</i>	A Python dictionary of {int : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.292 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u16string
> (
    PyObject * op,
    std::unordered_map< long, std::u16string > & map )
```

Instantiation for converting a Python dictionary {int : str, ...} to a C++ std::unordered_map<long, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {int : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.293 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u32string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u32string
> (
    PyObject * op,
    std::unordered_map< long, std::u32string > & map )
```

Instantiation for converting a Python dictionary {int : str, ...} to a C++ std::unordered_map<long, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {int : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.294 py_dict_to_cpp_std_map_like< std::unordered_map, long, std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::vector<
char > > (
    PyObject * op,
    std::unordered_map< long, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {int : bytes, ...} to a C++ std::unordered_map<long, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {int : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<long, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.295 py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::complex<
double >, bool > (
```

```
PyObject * op,
std::unordered_map< std::complex< double >, bool > & map )
```

Instantiation for converting a Python dictionary {complex : bool, ...} to a C++ std::unordered_map<std::complex<double>, bool>.

Parameters

<i>op</i>	A Python dictionary of {complex : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::complex<double>, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.296 py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::complex<
double >, double > (
    PyObject * op,
    std::unordered_map< std::complex< double >, double > & map )
```

Instantiation for converting a Python dictionary {complex : float, ...} to a C++ std::unordered_map<std::complex<double>, double>.

Parameters

<i>op</i>	A Python dictionary of {complex : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::complex<double>, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.297 py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::complex<
double >, long > (
    PyObject * op,
    std::unordered_map< std::complex< double >, long > & map )
```

Instantiation for converting a Python dictionary {complex : int, ...} to a C++ std::unordered_map<std::complex<double>, long>.

Parameters

<i>op</i>	A Python dictionary of {complex : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::complex<double>, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.298 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, bool
> (
    PyObject * op,
    std::unordered_map< std::string, bool > & map )
```

Instantiation for converting a Python dictionary {str : bool, ...} to a C++ std::unordered_map<std::string, bool>.

Parameters

<i>op</i>	A Python dictionary of {str : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.299 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, double
> (
    PyObject * op,
    std::unordered_map< std::string, double > & map )
```

Instantiation for converting a Python dictionary {str : float, ...} to a C++ std::unordered_map<std::string, double>.

Parameters

<i>op</i>	A Python dictionary of {str : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.300 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, long
> (
    PyObject * op,
    std::unordered_map< std::string, long > & map )
```

Instantiation for converting a Python dictionary {str : int, ...} to a C++ std::unordered_map<std::string, long>.

Parameters

<i>op</i>	A Python dictionary of {str : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.301 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::complex< double > > (
    PyObject * op,
    std::unordered_map< std::string, std::complex< double > > & map )
```

Instantiation for converting a Python dictionary {str : complex, ...} to a C++ std::unordered_map<std::string, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {str : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.302 `py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::string > (
    PyObject * op,
    std::unordered_map< std::string, std::string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::string, std::string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::string, std::string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.303 `py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u16string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u16string > (
    PyObject * op,
    std::unordered_map< std::string, std::u16string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::string, std::u16string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::string, std::u16string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.304 `py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u32string > (
```

```
PyObject * op,
std::unordered_map< std::string, std::u32string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::string, std::u32string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, std::u32string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.305 py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::vector< char > > (
    PyObject * op,
    std::unordered_map< std::string, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {str : bytes, ...} to a C++ std::unordered_map<std::string, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {str : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::string, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.306 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
bool > (
    PyObject * op,
    std::unordered_map< std::u16string, bool > & map )
```

Instantiation for converting a Python dictionary {str : bool, ...} to a C++ std::unordered_map<std::u16string, bool>.

Parameters

<i>op</i>	A Python dictionary of {str : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.307 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
double > (
    PyObject * op,
    std::unordered_map< std::u16string, double > & map )
```

Instantiation for converting a Python dictionary {str : float, ...} to a C++ std::unordered_map<std::u16string, double>.

Parameters

<i>op</i>	A Python dictionary of {str : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.308 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
long > (
    PyObject * op,
    std::unordered_map< std::u16string, long > & map )
```

Instantiation for converting a Python dictionary {str : int, ...} to a C++ std::unordered_map<std::u16string, long>.

Parameters

<i>op</i>	A Python dictionary of {str : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.309 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
std::complex< double > > (
    PyObject * op,
    std::unordered_map< std::u16string, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {str : complex, ...} to a C++ std::unordered_map<std::u16string, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {str : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.310 py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
std::string > (
    PyObject * op,
    std::unordered_map< std::u16string, std::string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::u16string, std::string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.311 `py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u16string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
std::u16string > (
    PyObject * op,
    std::unordered_map< std::u16string, std::u16string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::u16string, std::u16string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u16string, std::u16string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.312 `py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
std::u32string > (
    PyObject * op,
    std::unordered_map< std::u16string, std::u32string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::u16string, std::u32string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u16string, std::u32string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.313 `py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::vector< char > >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string,
std::vector< char > > > (
```

```
PyObject * op,
std::unordered_map< std::u16string, std::vector< char >> & map )
```

Instantiation for converting a Python dictionary {str : bytes, ...} to a C++ std::unordered_map<std::u16string, std::vector<char>>.

Parameters

<i>op</i>	A Python dictionary of {str : bytes, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u16string, std::vector<char>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.314 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, bool >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
bool > (
    PyObject * op,
    std::unordered_map< std::u32string, bool > & map )
```

Instantiation for converting a Python dictionary {str : bool, ...} to a C++ std::unordered_map<std::u32string, bool>.

Parameters

<i>op</i>	A Python dictionary of {str : bool, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, bool> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.315 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
double > (
    PyObject * op,
    std::unordered_map< std::u32string, double > & map )
```

Instantiation for converting a Python dictionary {str : float, ...} to a C++ std::unordered_map<std::u32string, double>.

Parameters

<i>op</i>	A Python dictionary of {str : float, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, double> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.316 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
long > (
    PyObject * op,
    std::unordered_map< std::u32string, long > & map )
```

Instantiation for converting a Python dictionary {str : int, ...} to a C++ std::unordered_map<std::u32string, long>.

Parameters

<i>op</i>	A Python dictionary of {str : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.317 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
std::complex< double > > (
    PyObject * op,
    std::unordered_map< std::u32string, std::complex< double >> & map )
```

Instantiation for converting a Python dictionary {str : complex, ...} to a C++ std::unordered_map<std::u32string, std::complex<double>>.

Parameters

<i>op</i>	A Python dictionary of {str : complex, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, std::complex<double>> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.318 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
std::string > (
    PyObject * op,
    std::unordered_map< std::u32string, std::string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::u32string, std::string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, std::string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.319 py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u16string >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
std::u16string > (
    PyObject * op,
    std::unordered_map< std::u32string, std::u16string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ std::unordered_map<std::u32string, std::u16string>.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::u32string, std::u16string> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.320 `py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u32string >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
std::u32string > (
    PyObject * op,
    std::unordered_map< std::u32string, std::u32string > & map )
```

Instantiation for converting a Python dictionary {str : str, ...} to a C++ `std::unordered_map<std::u32string, std::u32string>`.

Parameters

<i>op</i>	A Python dictionary of {str : str, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u32string, std::u32string></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.321 `py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::vector< char > >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string,
std::vector< char > > (
    PyObject * op,
    std::unordered_map< std::u32string, std::vector< char > > & map )
```

Instantiation for converting a Python dictionary {str : bytes, ...} to a C++ `std::unordered_map<std::u32string, std::vector<char>>`.

Parameters

<i>op</i>	A Python dictionary of {str : bytes, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::u32string, std::vector<char>></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.322 `py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, bool >()`

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char
>, bool > (
```

```
PyObject * op,
std::unordered_map< std::vector< char >, bool > & map )
```

Instantiation for converting a Python dictionary {bytes : bool, ...} to a C++ `std::unordered_map<std::vector<char>, bool>`.

Parameters

<i>op</i>	A Python dictionary of {bytes : bool, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::vector<char>, bool></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.323 py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, double >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char
>, double > (
    PyObject * op,
    std::unordered_map< std::vector< char >, double > & map )
```

Instantiation for converting a Python dictionary {bytes : float, ...} to a C++ `std::unordered_map<std::vector<char>, double>`.

Parameters

<i>op</i>	A Python dictionary of {bytes : float, ...} as the input.
<i>map</i>	The C++ <code>std::unordered_map<std::vector<char>, double></code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.324 py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, long >()

```
template<>
int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char
>, long > (
    PyObject * op,
    std::unordered_map< std::vector< char >, long > & map )
```

Instantiation for converting a Python dictionary {bytes : int, ...} to a C++ `std::unordered_map<std::vector<char>, long>`.

Parameters

<i>op</i>	A Python dictionary of {bytes : int, ...} as the input.
<i>map</i>	The C++ std::unordered_map<std::vector<char>, long> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.325 py_float_check()

```
int Python_Cpp_Containers::py_float_check (
    PyObject * op )
```

Return non-zero if the given value is a Python float type.

Parameters

<i>op</i>	The Python object to check to be a float type.
-----------	--

Returns

Zero if not a Python float, non-zero if a Python float.

7.1.3.326 py_float_to_cpp_double()

```
double Python_Cpp_Containers::py_float_to_cpp_double (
    PyObject * op )
```

Converts a Python float to a C++ float. This asserts that the given value is a Python float. If asserts are enabled then this asserts that the argument is a Python double objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ float.

7.1.3.327 py_frozenset_check()

```
int Python_Cpp_Containers::py_frozenset_check (
    PyObject * op )
```

Return non-zero if the given value is a Python `frozenset`. This is a wrapper around `PyFrozenSet_Check`

Parameters

<i>op</i>	The Python object to check to be a <code>frozenset</code> .
-----------	---

Returns

Zero if not a Python `frozenset`, non-zero if a Python `frozenset`.

7.1.3.328 py_frozenset_to_cpp_std_unordered_set()

```
template<typename T >
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set (
    PyObject * op,
    std::unordered_set< T > & container )
```

Base declaration for converting a Python `frozenset` to a C++ `std::unordered_set`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>op</i>	The Python container to read from.
<i>container</i>	The C++ <code>std::unordered_set</code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.329 py_frozenset_to_cpp_std_unordered_set< bool >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< bool > (
    PyObject * op,
    std::unordered_set< bool > & container )
```

Instantiation for converting a Python `frozenset` of `bool` to a C++ `std::unordered_set<bool>`.

Parameters

<i>op</i>	Python input as a frozenset of bool.
<i>container</i>	C++ output as a <code>std::unordered_set<bool></code> .

Returns

0 on success, non-zero on failure.

7.1.3.330 py_frozenset_to_cpp_std_unordered_set< double >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< double > (
    PyObject * op,
    std::unordered_set< double > & container )
```

Instantiation for converting a Python frozenset of float to a C++ `std::unordered_set<double>`.

Parameters

<i>op</i>	Python input as a frozenset of float.
<i>container</i>	C++ output as a <code>std::unordered_set<double></code> .

Returns

0 on success, non-zero on failure.

7.1.3.331 py_frozenset_to_cpp_std_unordered_set< long >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< long > (
    PyObject * op,
    std::unordered_set< long > & container )
```

Instantiation for converting a Python frozenset of int to a C++ `std::unordered_set<long>`.

Parameters

<i>op</i>	Python input as a frozenset of int.
<i>container</i>	C++ output as a <code>std::unordered_set<long></code> .

Returns

0 on success, non-zero on failure.

7.1.3.332 py_frozenset_to_cpp_std_unordered_set< std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::complex< double > > (
    PyObject * op,
    std::unordered_set< std::complex< double >> & container )
```

Instantiation for converting a Python frozenset of complex to a C++ `std::unordered_set<std::complex<double>>`.

Parameters

<i>op</i>	Python input as a frozenset of complex.
<i>container</i>	C++ output as a <code>std::unordered_set<std::complex<double>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.333 py_frozenset_to_cpp_std_unordered_set< std::string >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::string > (
    PyObject * op,
    std::unordered_set< std::string > & container )
```

Instantiation for converting a Python frozenset of str to a C++ `std::unordered_set<std::string>`.

Parameters

<i>op</i>	Python input as a frozenset of str.
<i>container</i>	C++ output as a <code>std::unordered_set<std::string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.334 py_frozenset_to_cpp_std_unordered_set< std::u16string >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::u16string > (
```

```
PyObject * op,
std::unordered_set< std::u16string > & container )
```

Instantiation for converting a Python frozenset of str to a C++ `std::unordered_set<std::u16string>`.

Parameters

<i>op</i>	Python input as a frozenset of str.
<i>container</i>	C++ output as a <code>std::unordered_set<std::u16string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.335 py_frozenset_to_cpp_std_unordered_set< std::u32string >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::u32string > (
    PyObject * op,
    std::unordered_set< std::u32string > & container )
```

Instantiation for converting a Python frozenset of str to a C++ `std::unordered_set<std::u32string>`.

Parameters

<i>op</i>	Python input as a frozenset of str.
<i>container</i>	C++ output as a <code>std::unordered_set<std::u32string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.336 py_frozenset_to_cpp_std_unordered_set< std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::vector< char > > (
    PyObject * op,
    std::unordered_set< std::vector< char >> & container )
```

Instantiation for converting a Python frozenset of bytes to a C++ `std::unordered_set<std::vector<char>>`.

Parameters

<i>op</i>	Python input as a <code>frozenset</code> of bytes.
<i>container</i>	C++ output as a <code>std::unordered_set<std::vector<char>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.337 py_list_check()

```
int Python_Cpp_Containers::py_list_check (
    PyObject * op )
```

Return non-zero if the given value is a Python list. This is a wrapper around `PyList_Check`

Parameters

<i>op</i>	The Python object to check to be a list.
-----------	--

Returns

Zero if not a Python list, non-zero if a Python list.

7.1.3.338 py_list_get()

```
PyObject * Python_Cpp_Containers::py_list_get (
    PyObject * list_p,
    size_t pos )
```

Get a value from the list. This is a wrapper around `PyList_GET_ITEM` This is undefined if passed a non-list or the position is out of range.

Parameters

<i>list_↔_p</i>	The Python list
<i>pos</i>	Index into the list to get.
<i>op</i>	Value to set, the reference is stolen.

Returns

Value to get, the reference is borrowed.

7.1.3.339 py_list_len()

```
Py_ssize_t Python_Cpp_Containers::py_list_len (
    PyObject * op )
```

Returns the length of the Python `list`. This is a wrapper around `PyList_Size` This is undefined if `*op` is not a `list`.

Parameters

<i>op</i>	The Python <code>list</code> .
-----------	--------------------------------

Returns

Length.

7.1.3.340 py_list_new()

```
PyObject * Python_Cpp_Containers::py_list_new (
    size_t len )
```

Creates a new Python `list`. This is a wrapper around `PyList_New`

Parameters

<i>len</i>	Required length of the container.
------------	-----------------------------------

Returns

A new empty container of required length or `NULL` on failure.

7.1.3.341 py_list_set()

```
int Python_Cpp_Containers::py_list_set (
    PyObject * list_p,
    size_t pos,
    PyObject * op )
```

Set a value in the `list`. This is a wrapper around `PyList_SET_ITEM` This is undefined if passed a non-`list` or the position is out of range.

Parameters

<i>list_p</i>	The Python <code>list</code>
<i>pos</i>	Index into the list to set.
<i>op</i>	Value to set, the reference is stolen.

Returns

0 on success.

7.1.3.342 py_list_to_cpp_std_list_like() [1/2]

```
template<typename T >
int Python_Cpp_Containers::py_list_to_cpp_std_list_like (
    PyObject * op,
    std::list< T > & container )
```

Base declaration for converting a Python `list` to a C++ `std::list`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>op</i>	The Python container to read from.
<i>container</i>	The C++ <code>std::list</code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.343 py_list_to_cpp_std_list_like() [2/2]

```
template<typename T >
int Python_Cpp_Containers::py_list_to_cpp_std_list_like (
    PyObject * op,
    std::vector< T > & container )
```

Base declaration for converting a Python `list` to a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>op</i>	The Python container to read from.
<i>container</i>	The C++ <code>std::vector</code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.344 py_list_to_cpp_std_list_like< bool >() [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< bool > (
    PyObject * op,
    std::list< bool > & container )
```

Instantiation for converting a Python list of bool to a C++ std::list<bool>.

Parameters

<i>op</i>	Python input as a list of bool.
<i>container</i>	C++ output as a std::list<bool>.

Returns

0 on success, non-zero on failure.

7.1.3.345 py_list_to_cpp_std_list_like< bool >() [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< bool > (
    PyObject * op,
    std::vector< bool > & container )
```

Instantiation for converting a Python list of bool to a C++ std::vector<bool>.

Parameters

<i>op</i>	Python input as a list of bool.
<i>container</i>	C++ output as a std::vector<bool>.

Returns

0 on success, non-zero on failure.

7.1.3.346 py_list_to_cpp_std_list_like< CppCustomObject >()

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< CppCustomObject > (
```

```
PyObject * op,
std::vector< CppCustomObject > & container )
```

7.1.3.347 py_list_to_cpp_std_list_like< double >() [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< double > (
    PyObject * op,
    std::list< double > & container )
```

Instantiation for converting a Python list of float to a C++ std::list<double>.

Parameters

<i>op</i>	Python input as a list of float.
<i>container</i>	C++ output as a std::list<double>.

Returns

0 on success, non-zero on failure.

7.1.3.348 py_list_to_cpp_std_list_like< double >() [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< double > (
    PyObject * op,
    std::vector< double > & container )
```

Instantiation for converting a Python list of float to a C++ std::vector<double>.

Parameters

<i>op</i>	Python input as a list of float.
<i>container</i>	C++ output as a std::vector<double>.

Returns

0 on success, non-zero on failure.

7.1.3.349 py_list_to_cpp_std_list_like< long >() [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (
```

```
PyObject * op,
std::list< long > & container )
```

Instantiation for converting a Python list of int to a C++ `std::list<long>`.

Parameters

<i>op</i>	Python input as a list of int.
<i>container</i>	C++ output as a <code>std::list<long></code> .

Returns

0 on success, non-zero on failure.

7.1.3.350 `py_list_to_cpp_std_list_like< long >()` [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (
    PyObject * op,
    std::vector< long > & container )
```

Instantiation for converting a Python list of int to a C++ `std::vector<long>`.

Parameters

<i>op</i>	Python input as a list of int.
<i>container</i>	C++ output as a <code>std::vector<long></code> .

Returns

0 on success, non-zero on failure.

7.1.3.351 `py_list_to_cpp_std_list_like< std::complex< double > >()` [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::complex< double > > (
    PyObject * op,
    std::list< std::complex< double >> & container )
```

Instantiation for converting a Python list of complex to a C++ `std::list<std::complex<double>>`.

Parameters

<i>op</i>	Python input as a list of complex.
<i>container</i>	C++ output as a <code>std::list<std::complex<double>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.352 py_list_to_cpp_std_list_like< std::complex< double > >() [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::complex< double > > (
    PyObject * op,
    std::vector< std::complex< double >> & container )
```

Instantiation for converting a Python list of complex to a C++ std::vector<std::complex<double>>.

Parameters

<i>op</i>	Python input as a list of complex.
<i>container</i>	C++ output as a std::vector<std::complex<double>>.

Returns

0 on success, non-zero on failure.

7.1.3.353 py_list_to_cpp_std_list_like< std::string >() [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::string > (
    PyObject * op,
    std::list< std::string > & container )
```

Instantiation for converting a Python list of str to a C++ std::list<std::string>.

Parameters

<i>op</i>	Python input as a list of str.
<i>container</i>	C++ output as a std::list<std::string>.

Returns

0 on success, non-zero on failure.

7.1.3.354 py_list_to_cpp_std_list_like< std::string >() [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::string > (
```

```
PyObject * op,
std::vector< std::string > & container )
```

Instantiation for converting a Python list of str to a C++ `std::vector<std::string>`.

Parameters

<i>op</i>	Python input as a list of str.
<i>container</i>	C++ output as a <code>std::vector<std::string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.355 `py_list_to_cpp_std_list_like< std::u16string >()` [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u16string > (
    PyObject * op,
    std::list< std::u16string > & container )
```

Instantiation for converting a Python list of str to a C++ `std::list<std::u16string>`.

Parameters

<i>op</i>	Python input as a list of str.
<i>container</i>	C++ output as a <code>std::list<std::u16string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.356 `py_list_to_cpp_std_list_like< std::u16string >()` [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u16string > (
    PyObject * op,
    std::vector< std::u16string > & container )
```

Instantiation for converting a Python list of str to a C++ `std::vector<std::u16string>`.

Parameters

<i>op</i>	Python input as a list of str.
<i>container</i>	C++ output as a <code>std::vector<std::u16string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.357 py_list_to_cpp_std_list_like< std::u32string >() [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u32string > (
    PyObject * op,
    std::list< std::u32string > & container )
```

Instantiation for converting a Python list of str to a C++ std::list<std::u32string>.

Parameters

<i>op</i>	Python input as a list of str.
<i>container</i>	C++ output as a std::list<std::u32string>.

Returns

0 on success, non-zero on failure.

7.1.3.358 py_list_to_cpp_std_list_like< std::u32string >() [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u32string > (
    PyObject * op,
    std::vector< std::u32string > & container )
```

Instantiation for converting a Python list of str to a C++ std::vector<std::u32string>.

Parameters

<i>op</i>	Python input as a list of str.
<i>container</i>	C++ output as a std::vector<std::u32string>.

Returns

0 on success, non-zero on failure.

7.1.3.359 py_list_to_cpp_std_list_like< std::vector< char > >() [1/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::vector< char > > (
```

```
PyObject * op,
std::list< std::vector< char >> & container )
```

Instantiation for converting a Python list of bytes to a C++ `std::list<std::vector<char>>`.

Parameters

<i>op</i>	Python input as a list of bytes.
<i>container</i>	C++ output as a <code>std::list<std::vector<char>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.360 `py_list_to_cpp_std_list_like< std::vector< char >>()` [2/2]

```
template<>
int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::vector< char >> (
    PyObject * op,
    std::vector< std::vector< char >> & container )
```

Instantiation for converting a Python list of bytes to a C++ `std::vector<std::vector<char>>`.

Parameters

<i>op</i>	Python input as a list of bytes.
<i>container</i>	C++ output as a <code>std::vector<std::vector<char>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.361 `py_long_check()`

```
int Python_Cpp_Containers::py_long_check (
    PyObject * op )
```

Return non-zero if the given value is a Python long type.

Parameters

<i>op</i>	The Python object to check to be a long type.
-----------	---

Returns

Zero if not a Python `long`, non-zero if a Python `long`.

7.1.3.362 `py_long_to_cpp_long()`

```
long Python_Cpp_Containers::py_long_to_cpp_long (
    PyObject * op )
```

Converts a Python `long` to a C++ `long`. This asserts that the given value is a Python `long`. If asserts are enabled then this asserts that the argument is a Python `long` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `long`.

7.1.3.363 `py_set_check()`

```
int Python_Cpp_Containers::py_set_check (
    PyObject * op )
```

Return non-zero if the given value is a Python `set`. This is a wrapper around `PySet_Check`

Parameters

<i>op</i>	The Python object to check to be a <code>set</code> .
-----------	---

Returns

Zero if not a Python `set`, non-zero if a Python `set`.

7.1.3.364 `py_set_to_cpp_std_unordered_set()`

```
template<typename T >
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set (
    PyObject * op,
    std::unordered_set< T > & container )
```

Base declaration for converting a Python `set` to a C++ `std::unordered_set`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>op</i>	The Python container to read from.
<i>container</i>	The C++ <code>std::unordered_set</code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.365 `py_set_to_cpp_std_unordered_set< bool >()`

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< bool > (
    PyObject * op,
    std::unordered_set< bool > & container )
```

Instantiation for converting a Python set of `bool` to a C++ `std::unordered_set<bool>`.

Parameters

<i>op</i>	Python input as a set of <code>bool</code> .
<i>container</i>	C++ output as a <code>std::unordered_set<bool></code> .

Returns

0 on success, non-zero on failure.

7.1.3.366 `py_set_to_cpp_std_unordered_set< double >()`

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< double > (
    PyObject * op,
    std::unordered_set< double > & container )
```

Instantiation for converting a Python set of `float` to a C++ `std::unordered_set<double>`.

Parameters

<i>op</i>	Python input as a set of <code>float</code> .
<i>container</i>	C++ output as a <code>std::unordered_set<double></code> .

Returns

0 on success, non-zero on failure.

7.1.3.367 py_set_to_cpp_std_unordered_set< long >()

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< long > (
    PyObject * op,
    std::unordered_set< long > & container )
```

Instantiation for converting a Python set of int to a C++ std::unordered_set<long>.

Parameters

<i>op</i>	Python input as a set of int.
<i>container</i>	C++ output as a std::unordered_set<long>.

Returns

0 on success, non-zero on failure.

7.1.3.368 py_set_to_cpp_std_unordered_set< std::complex< double > >()

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::complex< double > > (
    PyObject * op,
    std::unordered_set< std::complex< double > > & container )
```

Instantiation for converting a Python set of complex to a C++ std::unordered_set<std::complex<double>>.

Parameters

<i>op</i>	Python input as a set of complex.
<i>container</i>	C++ output as a std::unordered_set<std::complex<double>>.

Returns

0 on success, non-zero on failure.

7.1.3.369 py_set_to_cpp_std_unordered_set< std::string >()

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::string > (
    PyObject * op,
    std::unordered_set< std::string > & container )
```

Instantiation for converting a Python set of str to a C++ std::unordered_set<std::string>.

Parameters

<i>op</i>	Python input as a set of str.
<i>container</i>	C++ output as a std::unordered_set<std::string>.

Returns

0 on success, non-zero on failure.

7.1.3.370 py_set_to_cpp_std_unordered_set< std::u16string >()

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::u16string > (
    PyObject * op,
    std::unordered_set< std::u16string > & container )
```

Instantiation for converting a Python set of str to a C++ std::unordered_set<std::u16string>.

Parameters

<i>op</i>	Python input as a set of str.
<i>container</i>	C++ output as a std::unordered_set<std::u16string>.

Returns

0 on success, non-zero on failure.

7.1.3.371 py_set_to_cpp_std_unordered_set< std::u32string >()

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::u32string > (
    PyObject * op,
    std::unordered_set< std::u32string > & container )
```

Instantiation for converting a Python set of str to a C++ std::unordered_set<std::u32string>.

Parameters

<i>op</i>	Python input as a set of str.
<i>container</i>	C++ output as a <code>std::unordered_set<std::u32string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.372 py_set_to_cpp_std_unordered_set< std::vector< char > >()

```
template<>
int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::vector< char > > (
    PyObject * op,
    std::unordered_set< std::vector< char >> & container )
```

Instantiation for converting a Python set of bytes to a C++ `std::unordered_set<std::vector<char>>`.

Parameters

<i>op</i>	Python input as a set of bytes.
<i>container</i>	C++ output as a <code>std::unordered_set<std::vector<char>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.373 py_tuple_check()

```
int Python_Cpp_Containers::py_tuple_check (
    PyObject * op )
```

Return non-zero if the given value is a Python tuple. This is a wrapper around `PyTuple_Check`

Parameters

<i>op</i>	The Python object to check to be a tuple.
-----------	---

Returns

Zero if not a Python tuple, non-zero if a Python tuple.

7.1.3.374 py_tuple_get()

```
PyObject * Python_Cpp_Containers::py_tuple_get (
    PyObject * tuple_p,
    size_t pos )
```

Get a value from the tuple. This is a wrapper around PyTuple_GET_ITEM This is undefined if passed a non-tuple or the position is out of range.

Parameters

<i>tuple_p</i>	The Python tuple
<i>pos</i>	Index into the tuple to get.
<i>op</i>	Value to set, the reference is stolen.

Returns

Value to get, the reference is borrowed.

7.1.3.375 py_tuple_len()

```
Py_ssize_t Python_Cpp_Containers::py_tuple_len (
    PyObject * op )
```

Returns the length of the Python tuple. This is a wrapper around PyTuple_Size This is undefined if *op is not a tuple.

Parameters

<i>op</i>	The Python tuple.
-----------	-------------------

Returns

Length.

7.1.3.376 py_tuple_new()

```
PyObject * Python_Cpp_Containers::py_tuple_new (
    size_t len )
```

Creates a new Python tuple. This is a wrapper around PyTuple_New

Parameters

<i>len</i>	Required length of the container.
------------	-----------------------------------

Returns

A new empty container of required length or `NULL` on failure.

7.1.3.377 `py_tuple_set()`

```
int Python_Cpp_Containers::py_tuple_set (
    PyObject * tuple_p,
    size_t pos,
    PyObject * op )
```

Set a value in the tuple. This is a wrapper around `PyTuple_SET_ITEM` This is undefined if passed a non-tuple or the position is out of range.

Parameters

<i>tuple_p</i>	The Python tuple
<i>pos</i>	Index into the tuple to set.
<i>op</i>	Value to set, the reference is stolen.

Returns

0 on success.

7.1.3.378 `py_tuple_to_cpp_std_list_like()` [1/2]

```
template<typename T >
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like (
    PyObject * op,
    std::list< T > & container )
```

Base declaration for converting a Python tuple to a C++ `std::list`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

Parameters

<i>op</i>	The Python container to read from.
<i>container</i>	The C++ <code>std::list</code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.379 `py_tuple_to_cpp_std_list_like()` [2/2]

```
template<typename T >
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like (
    PyObject * op,
    std::vector< T > & container )
```

Base declaration for converting a Python `tuple` to a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type.
----------	-----------

Parameters

<i>op</i>	The Python container to read from.
<i>container</i>	The C++ <code>std::vector</code> to write to.

Returns

0 on success, non-zero on failure.

7.1.3.380 `py_tuple_to_cpp_std_list_like< bool >()` [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< bool > (
    PyObject * op,
    std::list< bool > & container )
```

Instantiation for converting a Python `tuple` of `bool` to a C++ `std::list<bool>`.

Parameters

<i>op</i>	Python input as a tuple of bool.
<i>container</i>	C++ output as a <code>std::list<bool></code> .

Returns

0 on success, non-zero on failure.

7.1.3.381 py_tuple_to_cpp_std_list_like< bool >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< bool > (
    PyObject * op,
    std::vector< bool > & container )
```

Instantiation for converting a Python tuple of bool to a C++ `std::vector<bool>`.

Parameters

<i>op</i>	Python input as a tuple of bool.
<i>container</i>	C++ output as a <code>std::vector<bool></code> .

Returns

0 on success, non-zero on failure.

7.1.3.382 py_tuple_to_cpp_std_list_like< double >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< double > (
    PyObject * op,
    std::list< double > & container )
```

Instantiation for converting a Python tuple of float to a C++ `std::list<double>`.

Parameters

<i>op</i>	Python input as a tuple of float.
<i>container</i>	C++ output as a <code>std::list<double></code> .

Returns

0 on success, non-zero on failure.

7.1.3.383 py_tuple_to_cpp_std_list_like< double >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< double > (
    PyObject * op,
    std::vector< double > & container )
```

Instantiation for converting a Python tuple of float to a C++ std::vector<double>.

Parameters

<i>op</i>	Python input as a tuple of float.
<i>container</i>	C++ output as a std::vector<double>.

Returns

0 on success, non-zero on failure.

7.1.3.384 py_tuple_to_cpp_std_list_like< long >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< long > (
    PyObject * op,
    std::list< long > & container )
```

Instantiation for converting a Python tuple of int to a C++ std::list<long>.

Parameters

<i>op</i>	Python input as a tuple of int.
<i>container</i>	C++ output as a std::list<long>.

Returns

0 on success, non-zero on failure.

7.1.3.385 py_tuple_to_cpp_std_list_like< long >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< long > (
    PyObject * op,
    std::vector< long > & container )
```

Instantiation for converting a Python tuple of int to a C++ std::vector<long>.

Parameters

<i>op</i>	Python input as a tuple of int.
<i>container</i>	C++ output as a <code>std::vector<long></code> .

Returns

0 on success, non-zero on failure.

7.1.3.386 py_tuple_to_cpp_std_list_like< std::complex< double > >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::complex< double > > (
    PyObject * op,
    std::list< std::complex< double >> & container )
```

Instantiation for converting a Python tuple of complex to a C++ `std::list<std::complex<double>>`.

Parameters

<i>op</i>	Python input as a tuple of complex.
<i>container</i>	C++ output as a <code>std::list<std::complex<double>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.387 py_tuple_to_cpp_std_list_like< std::complex< double > >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::complex< double > > (
    PyObject * op,
    std::vector< std::complex< double >> & container )
```

Instantiation for converting a Python tuple of complex to a C++ `std::vector<std::complex<double>>`.

Parameters

<i>op</i>	Python input as a tuple of complex.
<i>container</i>	C++ output as a <code>std::vector<std::complex<double>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.388 py_tuple_to_cpp_std_list_like< std::string >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::string > (
    PyObject * op,
    std::list< std::string > & container )
```

Instantiation for converting a Python tuple of str to a C++ `std::list<std::string>`.

Parameters

<i>op</i>	Python input as a tuple of str.
<i>container</i>	C++ output as a <code>std::list<std::string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.389 py_tuple_to_cpp_std_list_like< std::string >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::string > (
    PyObject * op,
    std::vector< std::string > & container )
```

Instantiation for converting a Python tuple of str to a C++ `std::vector<std::string>`.

Parameters

<i>op</i>	Python input as a tuple of str.
<i>container</i>	C++ output as a <code>std::vector<std::string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.390 py_tuple_to_cpp_std_list_like< std::u16string >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u16string > (
    PyObject * op,
    std::list< std::u16string > & container )
```

Instantiation for converting a Python tuple of str to a C++ `std::list<std::u16string>`.

Parameters

<i>op</i>	Python input as a tuple of str.
<i>container</i>	C++ output as a <code>std::list<std::u16string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.391 py_tuple_to_cpp_std_list_like< std::u16string >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u16string > (
    PyObject * op,
    std::vector< std::u16string > & container )
```

Instantiation for converting a Python tuple of str to a C++ `std::vector<std::u16string>`.

Parameters

<i>op</i>	Python input as a tuple of str.
<i>container</i>	C++ output as a <code>std::vector<std::u16string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.392 py_tuple_to_cpp_std_list_like< std::u32string >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u32string > (
    PyObject * op,
    std::list< std::u32string > & container )
```

Instantiation for converting a Python tuple of str to a C++ `std::list<std::u32string>`.

Parameters

<i>op</i>	Python input as a tuple of str.
<i>container</i>	C++ output as a <code>std::list<std::u32string></code> .

Returns

0 on success, non-zero on failure.

7.1.3.393 py_tuple_to_cpp_std_list_like< std::u32string >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u32string > (
    PyObject * op,
    std::vector< std::u32string > & container )
```

Instantiation for converting a Python tuple of str to a C++ std::vector<std::u32string>.

Parameters

<i>op</i>	Python input as a tuple of str.
<i>container</i>	C++ output as a std::vector<std::u32string>.

Returns

0 on success, non-zero on failure.

7.1.3.394 py_tuple_to_cpp_std_list_like< std::vector< char > >() [1/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::vector< char > > (
    PyObject * op,
    std::list< std::vector< char >> & container )
```

Instantiation for converting a Python tuple of bytes to a C++ std::list<std::vector<char>>.

Parameters

<i>op</i>	Python input as a tuple of bytes.
<i>container</i>	C++ output as a std::list<std::vector<char>>.

Returns

0 on success, non-zero on failure.

7.1.3.395 py_tuple_to_cpp_std_list_like< std::vector< char > >() [2/2]

```
template<>
int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::vector< char > > (
    PyObject * op,
    std::vector< std::vector< char >> & container )
```

Instantiation for converting a Python tuple of bytes to a C++ std::vector<std::vector<char>>.

Parameters

<i>op</i>	Python input as a tuple of bytes.
<i>container</i>	C++ output as a <code>std::vector<std::vector<char>></code> .

Returns

0 on success, non-zero on failure.

7.1.3.396 py_unicode16_check()

```
int Python_Cpp_Containers::py_unicode16_check (
    PyObject * op )
```

Returns 1 if the Python object is a 16 bit Unicode string.

Return non-zero if the given value is a Python `str` type with 16 bit Unicode.

Parameters

<i>op</i>	The Python object to check to be a <code>str</code> type with 16 bit unicode.
-----------	---

Returns

Zero if not a Python `str`, non-zero if a Python `str`.

7.1.3.397 py_unicode16_to_cpp_u16string()

```
std::u16string Python_Cpp_Containers::py_unicode16_to_cpp_u16string (
    PyObject * op )
```

Converts a Python 16 bit Unicode string to a C++ `std::u16string`.

Converts a Python `str` to a C++ `std::u16string`. This asserts that the given value is a Python `str` with `PyUnicode_2BYTE_KIND` entries. If asserts are enabled then this asserts that the argument is a Python `str` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `std::str`.

7.1.3.398 py_unicode32_check()

```
int Python_Cpp_Containers::py_unicode32_check (
    PyObject * op )
```

Returns 1 if the Python object is a 32 bit Unicode string.

Return non-zero if the given value is a Python `str` type with 32 bit Unicode.

Parameters

<i>op</i>	The Python object to check to be a <code>str</code> type with 32 bit unicode.
-----------	---

Returns

Zero if not a Python `str`, non-zero if a Python `str`.

7.1.3.399 py_unicode32_to_cpp_u32string()

```
std::u32string Python_Cpp_Containers::py_unicode32_to_cpp_u32string (
    PyObject * op )
```

Converts a Python 32 bit Unicode string to a C++ `std::u32string`.

Converts a Python `str` to a C++ `std::u32string`. This asserts that the given value is a Python `str` with `PyUnicode_2BYTE_KIND` entries. If asserts are enabled then this asserts that the argument is a Python `str` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `std::str`.

7.1.3.400 py_unicode8_check()

```
int Python_Cpp_Containers::py_unicode8_check (
    PyObject * op )
```

Returns 1 if the Python object is a 8 bit Unicode string.

Return non-zero if the given value is a Python `str` type with `PyUnicode_1BYTE_KIND` entries.

Parameters

<i>op</i>	The Python object to check to be a <code>str</code> type with <code>PyUnicode_1BYTE_KIND</code> entries.
-----------	--

Returns

Zero if not a Python `str`, non-zero if a Python `str`.

7.1.3.401 `py_unicode8_to_cpp_string()`

```
std::string Python_Cpp_Containers::py_unicode8_to_cpp_string (
    PyObject * op )
```

Converts a Python 8 bit Unicode string to a C++ `std::string`.

Converts a Python `str` to a C++ `std::string`. This asserts that the given value is a Python `str`. If asserts are enabled then this asserts that the argument is a Python `str` objects. If asserts are not enabled then this is undefined.

Parameters

<i>op</i>	Python value to convert.
-----------	--------------------------

Returns

The C++ `std::str`.

7.1.3.402 `very_generic_cpp_std_list_like_to_py_unary()`

```
template<template< typename ... > class ListLike, typename T , PyObject *(*)(const T &)
ConvertCppToPy, PyObject *(*)(size_t) PyUnaryContainer_New, int(*)(PyObject *, size_t, PyObject
*) PyUnaryContainer_Set>
PyObject* Python_Cpp_Containers::very_generic_cpp_std_list_like_to_py_unary (
    const ListLike< T > & list_like )
```

This is a hand written generic function to convert a C++ `std::vector` to a Python tuple or list. The template is instantiated with a C++ type and a conversion function to create a Python object from that type.

Example of a partial specialisation of this template to create a function that will convert a C++ `std::vector<T>` to a Python tuple of T:

```
template<typename T, PyObject *(*Convert)(const T &)>
PyObject *
generic_cpp_std_vector_to_py_tuple(const std::vector<T> &vec) {
    return generic_cpp_std_vector_to_py_unary<T, Convert, &py_tuple_new, &py_tuple_set>(vec);
}
```

Example of a complete instantiation of this template to create a function that will convert a C++ `std::vector<double>` to a Python tuple of float. We have to supply the conversion function, a function to create a new tuple and a function to set a value:

```
PyObject *std_vector_to_py_tuple(const std::vector<double> &container) {
    return generic_cpp_std_vector_to_py_tuple<double, &cpp_double_to_py_float, &py_tuple_new,
        &py_tuple_set>(container);
}
```

This is a partial template specialisation. For complete specialisation we need `PyTuple_New`, `PyTuple_GET_ITEM`, `PyTuple_SET_ITEM`.

Template Parameters

<i>T</i>	The C++ type of the objects in the vector.
<i>ConvertCppToPy</i>	Function to convert the C++ T to a <code>PyObject*</code> .
<i>PyUnaryContainer_New</i>	Function to create a new Python container of a given length given as a <code>size_t</code> .
<i>PyUnaryContainer_Set</i>	Function to set a value in a Python container at a given position given as a <code>size_t</code> .

Parameters

<i>vec</i>	The C++ vector as input data.
------------	-------------------------------

Returns

The `PyObject*` as the output data containing the values of the C++ `std::vector` or NULL on failure in which case a `PyErr...` will be set.

7.1.3.403 very_generic_py_unary_to_cpp_std_list_like()

```
template<template< typename ... > class ListLike, typename T , int(*) (PyObject *) PyObject_↵
Check, T(*) (PyObject *) PyObject_Convert, int(*) (PyObject *) PyUnaryContainer_Check, Py_ssize_↵
_t(*) (PyObject *) PyUnaryContainer_Size, PyObject (*)(PyObject *, size_t) PyUnaryContainer_↵
Get>
int Python_Cpp_Containers::very_generic_py_unary_to_cpp_std_list_like (
    PyObject * op,
    ListLike< T > & list_like )
```

This is a hand written generic function to convert a Python tuple or list to a C++ `std::vector`. The template is instantiated with a C++ type a check function and a conversion function to create a Python object to that C++ type.

The given vector is cleared whether an error condition exists or not.

Error handling notes:

An assertion is made that no `PyErr` exists. If `PyUnary_Check(op)` then this sets a Python `ValueError`.

Example of an instantiation of this template to create a function that will convert a Python tuple of float to a C++ `std::vector<double>`:

```
template <> int
py_tuple_to_std_vector<double>(PyObject *op, std::vector<double> &container) {
    return generic_py_tuple_to_cpp_std_vector<double, &py_float_check, &py_float_to_cpp_double>(op,
        container);
}
```

This is a partial template specialisation. For complete specialisation we need `PyTuple_Check`, `PyTuple_↵`
`GET_SIZE`, `PyTuple_GET_ITEM`.

Template Parameters

<i>T</i>	The C++ type of the objects in the vector.
<i>PyObject_Check</i>	A function that takes a <code>PyObject*</code> and returns 1 if it is of the right type, 0 otherwise.
<i>PyObject_Convert</i>	A function to convert a <code>PyObject*</code> to a C++ <code>T</code> .
<i>PyUnaryContainer_Check</i>	A function that takes a <code>PyObject*</code> and returns 1 if it is of a suitable container, 0 otherwise.
<i>PyUnaryContainer_Size</i>	A function that returns the length of the Python container.
<i>PyUnaryContainer_Get</i>	A function that gets a <code>PyObject*</code> from the Python container at a given index as a <code>size_t</code> .

Parameters

<i>op</i>	The Python container of values that can be converted to C++ type <code>T</code> .
<i>vec</i>	The C++ <code>std::vector</code> to populate. This will be empty on failure.

Returns

0 on success, non-zero on failure.

7.2 src Namespace Reference

Namespaces

- [py](#)

7.3 src.py Namespace Reference

Namespaces

- [code_gen](#)
- [code_gen_common](#)
- [code_gen_documentation](#)

7.4 src.py.code_gen Namespace Reference

Classes

- class [CodeCount](#)

Functions

- str `defn_name_from_decl_name` (str name, str cpp_container)
- `CodeCount` unary_declarations ()
- `CodeCount` unary_definitions ()
- `CodeCount` dict_map_declarations ()
- `CodeCount` dict_map_definitions ()
- `CodeCount` declarations ()
- `CodeCount` definitions ()
- None `write_files` ()
- def `main` ()

Variables

- `logger` = logging.getLogger(__file__)
- string `CPP_NAMESPACE` = 'Python_Cpp_Containers'
- string `PROJECT_VERSION` = '0.4.0'
- dictionary `CPP_TYPE_TO_FUNCS`
- tuple `UNARY_COLLECTIONS`
- list `REQUIRED_INCLUDES`
- dictionary `CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINER`
- string `CPP_UNARY_FUNCTION_TO_PY_BASE_DECL`
- string `CPP_UNARY_FUNCTION_TO_PY_DECL`
- string `CPP_UNARY_FUNCTION_TO_PY_DEFN`
- string `PY_TO_CPP_UNARY_FUNCTION_BASE_DECL`
- string `PY_TO_CPP_UNARY_FUNCTION_DECL`
- string `PY_TO_CPP_UNARY_FUNCTION_DEFN`
- tuple `CPP_MAP_TYPES` = ('std::unordered_map', 'std::map')
- string `CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL`
- string `CPP_MAP_TYPE_TO_PY_DICT_DECL`
- string `CPP_MAP_TYPE_TO_PY_DICT_DEFN`
- string `CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL`
- string `CPP_PY_DICT_TO_MAP_TYPE_DECL`
- string `CPP_PY_DICT_TO_MAP_TYPE_DEFN`
- string `AUTO_FILE_NAME` = 'auto_py_convert_internal'

7.4.1 Detailed Description

Writes out .h and .cpp files to support Python/C++ homogeneous containers.

This facilitates conversion between Python and C++ containers where the Python types are consistent.

For example a Python set of strings to and from a C++ unordered_set<std::string>

Note on nomenclature:

- 'cpp' is C++
- C++ namespaced types are '_' separated so 'std::vector' is 'cpp_std_vector'
- 'py' is Python
- Conversion functions are always ..._to_...

7.4.2 Function Documentation

7.4.2.1 declarations()

`CodeCount` src.py.code_gen.declarations ()

Returns the C++ code for all declarations.

7.4.2.2 definitions()

`CodeCount` src.py.code_gen.definitions ()

Returns the C++ code for all definitions.

7.4.2.3 defn_name_from_decl_name()

```
str src.py.code_gen.defn_name_from_decl_name (
    str name,
    str cpp_container )
```

Returns the definition name given the declaration name by the convention that it is preceded with 'generic_'. These 'generic_*' functions are handwritten templates in python_convert.h

7.4.2.4 dict_map_declarations()

`CodeCount` src.py.code_gen.dict_map_declarations ()

Returns the C++ code for the Python dictionary declarations.

7.4.2.5 dict_map_definitions()

`CodeCount` src.py.code_gen.dict_map_definitions ()

Returns the C++ code for the Python dictionary definitions.

7.4.2.6 main()

```
def src.py.code_gen.main ( )
```

7.4.2.7 unary_declarations()

```
CodeCount src.py.code_gen.unary_declarations ( )
```

Returns the C++ code for the unary declarations (tuples, lists, sets and so on).

7.4.2.8 unary_definitions()

```
CodeCount src.py.code_gen.unary_definitions ( )
```

Returns the C++ code for the unary definitions (tuples, lists, sets and so on).

7.4.2.9 write_files()

```
None src.py.code_gen.write_files ( )
```

Writes all C++ files.

7.4.3 Variable Documentation

7.4.3.1 AUTO_FILE_NAME

```
string src.py.code_gen.AUTO_FILE_NAME = 'auto_py_convert_internal'
```

7.4.3.2 CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL

```
string src.py.code_gen.CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL
```

Initial value:

```
1 = """template<typename ...> class Map, typename K, typename V>  
2 PyObject *  
3 cpp_std_map_like_to_py_dict(const Map<K, V> &map);"""
```


7.4.3.3 CPP_MAP_TYPE_TO_PY_DICT_DECL

```
string src.py.code_gen.CPP_MAP_TYPE_TO_PY_DICT_DECL
```

Initial value:

```
1 = """template <>
2 PyObject *
3 cpp_std_map_like_to_py_dict<{cpp_map_type}, {cpp_type_K}, {cpp_type_V}>(const {cpp_map_type}<{cpp_type_K},
    {cpp_type_V}> &map);"""
```

7.4.3.4 CPP_MAP_TYPE_TO_PY_DICT_DEFN

```
string src.py.code_gen.CPP_MAP_TYPE_TO_PY_DICT_DEFN
```

Initial value:

```
1 = """template <>
2 PyObject *
3 cpp_std_map_like_to_py_dict<{cpp_map_type}, {type_K}, {type_V}>(const {cpp_map_type}<{type_K}, {type_V}>
    &map) {{
4     return generic_cpp_std_map_like_to_py_dict<
5         {cpp_map_type},
6         {type_K}, {type_V},
7         &{convert_K_to_py}, &{convert_V_to_py}
8     >(&map);
9 }}
10 """
```

7.4.3.5 CPP_MAP_TYPES

```
tuple src.py.code_gen.CPP_MAP_TYPES = ('std::unordered_map', 'std::map')
```

7.4.3.6 CPP_NAMESPACE

```
string src.py.code_gen.CPP_NAMESPACE = 'Python_Cpp_Containers'
```

7.4.3.7 CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL

```
string src.py.code_gen.CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL
```

Initial value:

```
1 = """template<typename ...> class Map, typename K, typename V>
2 int
3 py_dict_to_cpp_std_map_like(PyObject *op, Map<K, V> &map);"""
```

7.4.3.8 CPP_PY_DICT_TO_MAP_TYPE_DECL

```
string src.py.code_gen.CPP_PY_DICT_TO_MAP_TYPE_DECL
```

Initial value:

```
1 = """template <>
2 int
3 py_dict_to_cpp_std_map_like<{cpp_map_type}, {cpp_type_K}, {cpp_type_V}>(PyObject* op,
  {cpp_map_type}<{cpp_type_K}, {cpp_type_V}> &map);"""
```

7.4.3.9 CPP_PY_DICT_TO_MAP_TYPE_DEFN

```
string src.py.code_gen.CPP_PY_DICT_TO_MAP_TYPE_DEFN
```

Initial value:

```
1 = """template <>
2 int
3 py_dict_to_cpp_std_map_like<{cpp_map_type}, {type_K}, {type_V}>(PyObject* op, {cpp_map_type}<{type_K},
  {type_V}> &map) {{
4     return generic_py_dict_to_cpp_std_map_like<
5         {cpp_map_type},
6         {type_K}, {type_V},
7         &{py_check_K}, &{py_check_V},
8         &{convert_K_from_py}, &{convert_V_from_py}
9     >(op, map);
10 }}
11 """
```

7.4.3.10 CPP_TYPE_TO_FUNCS

```
dictionary src.py.code_gen.CPP_TYPE_TO_FUNCS
```

Initial value:

```
1 = {
2     'bool': code_gen_common.CppTypeFunctions('cpp_bool_to_py_bool', 'py_bool_check',
3         'py_bool_to_cpp_bool', 'bool'),
4     'long': code_gen_common.CppTypeFunctions('cpp_long_to_py_long', 'py_long_check',
5         'py_long_to_cpp_long', 'int'),
6     'double': code_gen_common.CppTypeFunctions('cpp_double_to_py_float', 'py_float_check',
7         'py_float_to_cpp_double',
8         'float'),
9     'std::complex<double>': code_gen_common.CppTypeFunctions('cpp_complex_to_py_complex',
10        'py_complex_check',
11        'py_complex_to_cpp_complex', 'complex'),
12     'std::vector<char>': code_gen_common.CppTypeFunctions('cpp_vector_char_to_py_bytes', 'py_bytes_check',
13        'py_bytes_to_cpp_vector_char', 'bytes'),
14     'std::string': code_gen_common.CppTypeFunctions('cpp_string_to_py_unicode8', 'py_unicode8_check',
15        'py_unicode8_to_cpp_string', 'str'),
16     'std::u16string': code_gen_common.CppTypeFunctions('cpp_u16string_to_py_unicode16',
17        'py_unicode16_check',
18        'py_unicode16_to_cpp_u16string', 'str'),
19     'std::u32string': code_gen_common.CppTypeFunctions('cpp_u32string_to_py_unicode32',
20        'py_unicode32_check',
21        'py_unicode32_to_cpp_u32string', 'str'),
22 }
```

7.4.3.11 CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINER

```
dictionary src.py.code_gen.CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINER
```

Initial value:

```
1 = {
2     # Example: 'std::unordered_map' : 'MyClass',
3 }
```

7.4.3.12 CPP_UNARY_FUNCTION_TO_PY_BASE_DECL

```
string src.py.code_gen.CPP_UNARY_FUNCTION_TO_PY_BASE_DECL
```

Initial value:

```
1 = """template<typename T>
2 PyObject *
3 {fn}(const {cpp_container}<T> &container);"""
```

7.4.3.13 CPP_UNARY_FUNCTION_TO_PY_DECL

```
string src.py.code_gen.CPP_UNARY_FUNCTION_TO_PY_DECL
```

Initial value:

```
1 = """template <>
2 PyObject *
3 {fn}<{cpp_type}>(const {cpp_container}<{cpp_type}> &container);"""
```

7.4.3.14 CPP_UNARY_FUNCTION_TO_PY_DEFN

```
string src.py.code_gen.CPP_UNARY_FUNCTION_TO_PY_DEFN
```

Initial value:

```
1 = """template <>
2 PyObject *
3 {fn_decl}<{cpp_type}>(const {cpp_container}<{cpp_type}> &container) {{
4     return {fn_defn}<{cpp_type}>, &{convert_to_py}>(container);
5 }}
6 """
```

7.4.3.15 logger

```
src.py.code_gen.logger = logging.getLogger(__file__)
```

7.4.3.16 PROJECT_VERSION

```
string src.py.code_gen.PROJECT_VERSION = '0.4.0'
```

7.4.3.17 PY_TO_CPP_UNARY_FUNCTION_BASE_DECL

```
string src.py.code_gen.PY_TO_CPP_UNARY_FUNCTION_BASE_DECL
```

Initial value:

```
1 = """template<typename T>
2 int
3 {fn}(PyObject *op, {cpp_container}<T> &container);"""
```

7.4.3.18 PY_TO_CPP_UNARY_FUNCTION_DECL

```
string src.py.code_gen.PY_TO_CPP_UNARY_FUNCTION_DECL
```

Initial value:

```
1 = """template <>
2 int
3 {fn}<{cpp_type}>(PyObject *op, {cpp_container}<{cpp_type}> &container);"""
```

7.4.3.19 PY_TO_CPP_UNARY_FUNCTION_DEFN

```
string src.py.code_gen.PY_TO_CPP_UNARY_FUNCTION_DEFN
```

Initial value:

```
1 = """template <>
2 int
3 {fn_decl}<{cpp_type}>(PyObject *op, {cpp_container}<{cpp_type}> &container) {{
4     return {fn_defn}<{cpp_type}>, &{py_check}, &{convert_from_py}>(op, container);
5 }}
6 """
```

7.4.3.20 REQUIRED_INCLUDES

```
list src.py.code_gen.REQUIRED_INCLUDES
```

Initial value:

```
1 = [
2     # Example: 'foo.h'
3 ]
```

7.4.3.21 UNARY_COLLECTIONS

tuple src.py.code_gen.UNARY_COLLECTIONS

Initial value:

```
1 = (
2     code_gen_common.UnaryFunctions('tuple', 'std::vector', 'cpp_std_list_like_to_py_tuple',
3                                     'py_tuple_to_cpp_std_list_like'),
4     code_gen_common.UnaryFunctions('tuple', 'std::list', 'cpp_std_list_like_to_py_tuple',
5                                     'py_tuple_to_cpp_std_list_like'),
6     code_gen_common.UnaryFunctions('list', 'std::vector', 'cpp_std_list_like_to_py_list',
7                                     'py_list_to_cpp_std_list_like'),
8     code_gen_common.UnaryFunctions('list', 'std::list', 'cpp_std_list_like_to_py_list',
9                                     'py_list_to_cpp_std_list_like'),
9     code_gen_common.UnaryFunctions('set', 'std::unordered_set', 'cpp_std_unordered_set_to_py_set',
10                                    'py_set_to_cpp_std_unordered_set'),
11     code_gen_common.UnaryFunctions('frozenset', 'std::unordered_set',
12                                    'cpp_std_unordered_set_to_py_frozenset',
13                                    'py_frozenset_to_cpp_std_unordered_set'),
14 )
```

7.5 src.py.code_gen_common Namespace Reference

Classes

- class [CppTypeFunctions](#)
- class [TypeConversionFunctions](#)
- class [UnaryFunctions](#)

7.6 src.py.code_gen_documentation Namespace Reference

Functions

- typing.List[str] [doxygen_cpp_to_python_unary_base_class](#) (str cpp_container, str python_container)
- def [doxygen_cpp_to_python_unary_instantiation](#) (str cpp_container, str python_container, str cpp_type, str py_type)
- typing.List[str] [doxygen_python_to_cpp_unary_base_class](#) (str cpp_container, str python_container)
- def [doxygen_python_to_cpp_unary_instantiation](#) (str cpp_container, str python_container, str cpp_type, str py_type)
- typing.List[str] [doxygen_cpp_to_python_dict_base_class](#) ()
- def [doxygen_cpp_to_python_dict_instantiation](#) (str cpp_key_type, str cpp_val_type, str py_key_type, str py↔_val_type)
- def [doxygen_python_dict_to_cpp_base_class](#) ()
- def [doxygen_python_dict_to_cpp_instantiation](#) (str cpp_key_type, str cpp_val_type, str py_key_type, str py↔_val_type)
- str [comment_str](#) (str s)
- typing.List[str] [comment_list_str](#) (typing.List[str] inputs)
- def [cpp_comment_section](#) (typing.List[str] str_list, str title, str sep)
- typing.List[str] [documentation](#) (typing.Tuple[[code_gen_common.UnaryFunctions](#),...] unary_collections, typing.Dict[str, [code_gen_common.CppTypeFunctions](#)] cpp_type_to_funcs)
- typing.List[str] [get_codegen_please_no_edit_warning](#) (bool is_end)
- def [get_codegen_please_no_edit_warning_context](#) (typing.List[str] str_list)

Variables

- int [WIDTH](#) = 75 - len('//')

7.6.1 Detailed Description

Provides Doxygen style comments for code_gen.py

7.6.2 Function Documentation

7.6.2.1 comment_list_str()

```
typing.List[str] src.py.code_gen_documentation.comment_list_str (  
    typing.List[str] inputs )
```

Returns the strings as a C++ comments.

7.6.2.2 comment_str()

```
str src.py.code_gen_documentation.comment_str (  
    str s )
```

Turn a single line string into an inline comment.

7.6.2.3 cpp_comment_section()

```
def src.py.code_gen_documentation.cpp_comment_section (  
    typing.List[str] str_list,  
    str title,  
    str sep )
```

Context manager for writing beginning and end comments.

7.6.2.4 documentation()

```
typing.List[str] src.py.code_gen_documentation.documentation (  
    typing.Tuple[code\_gen\_common.UnaryFunctions, ...] unary_collections,  
    typing.Dict[str, code\_gen\_common.CppTypeFunctions] cpp_type_to_funcs )
```

General documentation.

7.6.2.5 doxygen_cpp_to_python_dict_base_class()

```
typing.List[str] src.py.code_gen_documentation.doxygen_cpp_to_python_dict_base_class ( )

/**
 * * Base declaration for converting a C++ std::unordered_map<K, V> to a Python dictionary.
 *
 * @tparam K The C++ type for the key.
 * @tparam V The C++ type for the value.
 * @param map Input C++ std::unordered_map<K, V>.
 * @return Python dictionary corresponding to {K : V, ...}.
 */

For::
    template<typename K, typename V>
    PyObject *
    cpp_std_unordered_map_to_py_dict(const std::unordered_map<K, V> &map);
```

7.6.2.6 doxygen_cpp_to_python_dict_instantiation()

```
def src.py.code_gen_documentation.doxygen_cpp_to_python_dict_instantiation (
    str cpp_key_type,
    str cpp_val_type,
    str py_key_type,
    str py_val_type )

/**
 * Converts a C++ std::unordered_map<long, bool> to a Python dictionary.
 *
 * @param map Input C++ std::unordered_map<long, bool>.
 * @return A Python dictionary of {int : bool, ...}.
 */

For::

    template <>
    PyObject *
    cpp_std_unordered_map_to_py_dict<long, bool>(const std::unordered_map<long, bool> &map);
```

7.6.2.7 doxygen_cpp_to_python_unary_base_class()

```
typing.List[str] src.py.code_gen_documentation.doxygen_cpp_to_python_unary_base_class (
    str cpp_container,
    str python_container )
```

Returns a Doxygen style comment. For example::

```
/**
 * Base declaration for converting C++ vectors to a Python tuple.
 *
 * @tparam T C++ type.
 * @param container C++ input as a std::vector<T>.
 * @return A Python tuple containing type T.
 */
```

This is for::

```
template<typename T>
PyObject *
cpp_std_vector_to_py_tuple(const std::vector<T> &container);
```

7.6.2.8 doxygen_cpp_to_python_unary_instantiation()

```
def src.py.code_gen_documentation.doxygen_cpp_to_python_unary_instantiation (
    str cpp_container,
    str python_container,
    str cpp_type,
    str py_type )
```

Returns a Doxygen style comment. For example::

```
/**
 * Instantiation for converting C++ vectors of bool to a Python tuple of bool.
 *
 * @param container C++ input as a std::vector<bool>.
 * @return A Python tuple containing bool objects.
 */
```

This is for::

```
template <>
PyObject *
cpp_std_vector_to_py_tuple<bool>(const std::vector<bool> &container);
```

7.6.2.9 doxygen_python_dict_to_cpp_base_class()

```
def src.py.code_gen_documentation.doxygen_python_dict_to_cpp_base_class ( )
```

Example::

```
/**
 * Base declaration for converting a Python dictionary to a C++ std::unordered_map<K, V>.
 *
 * @tparam K The type of the C++ key.
 * @tparam V The type of the C++ value.
 * @param op The Python dictionary as the input.
 * @param map C++ std::unordered_map<K, V> to write to.
 * @return 0 on success, non-zero on failure.
 */
```

From::

```
template<typename K, typename V>
int
py_dict_to_cpp_std_unordered_map(PyObject *op, std::unordered_map<K, V> &map);
```

7.6.2.10 doxygen_python_dict_to_cpp_instantiation()

```
def src.py.code_gen_documentation.doxygen_python_dict_to_cpp_instantiation (
    str cpp_key_type,
    str cpp_val_type,
    str py_key_type,
    str py_val_type )
```


Example::

```
/**
 * Instantiation for converting a Python dictionary to a C++ std::unordered_map<long, bool>.
 *
 * @param op The Python dictionary as the input.
 * @param map C++ std::unordered_map<long, bool> to write to.
 * @return 0 on success, non-zero on failure.
 */
```

From::

```
template <>
int
py_dict_to_cpp_std_unordered_map<bool, bool>(PyObject* op, std::unordered_map<long, bool> &map);
```

7.6.2.11 doxygen_python_to_cpp_unary_base_class()

```
typing.List[str] src.py.code_gen_documentation.doxygen_python_to_cpp_unary_base_class (
    str cpp_container,
    str python_container )
```

Returns a Doxygen style comment. For example::

```
/**
 * Base declaration for converting a Python tuple to a C++ std::vector.
 *
 * @tparam T The C++ type.
 * @param op The Python container to read from.
 * @param container The C++ std::vector to write to.
 * @return 0 on success, non-zero on failure.
 */
```

This is for::

```
template<typename T>
int
py_tuple_to_cpp_std_vector(PyObject *op, std::vector<T> &container);
```

7.6.2.12 doxygen_python_to_cpp_unary_instantiation()

```
def src.py.code_gen_documentation.doxygen_python_to_cpp_unary_instantiation (
    str cpp_container,
    str python_container,
    str cpp_type,
    str py_type )
```

Returns a Doxygen style comment. For example::

```
/**
 * Instantiation for converting a Python tuple of bool to a C++ std::vector<bool>.
 *
 * @param op The Python container to read from.
 * @param container The C++ container to write to.
 * @return 0 on success, non-zero on failure.
 */
```

This is for::

```
template <>
int
py_tuple_to_cpp_std_vector<bool>(PyObject *op, std::vector<bool> &container);
```

7.6.2.13 `get_codegen_please_no_edit_warning()`

```
typing.List[str] src.py.code_gen_documentation.get_codegen_please_no_edit_warning (
    bool is_end )
```

Writes the start or end of a warning comment.

7.6.2.14 `get_codegen_please_no_edit_warning_context()`

```
def src.py.code_gen_documentation.get_codegen_please_no_edit_warning_context (
    typing.List[str] str_list )
```

Context manager that writes the start or end of a warning comment.

7.6.3 Variable Documentation

7.6.3.1 WIDTH

```
int src.py.code_gen_documentation.WIDTH = 75 - len('///  
'
```

7.7 std Namespace Reference

Classes

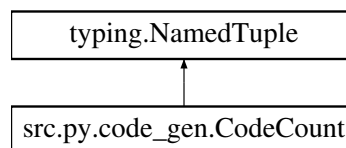
- struct [hash< std::vector< char > >](#)
- struct [hash< std::complex< double > >](#)
- struct [less< std::complex< T > >](#)

Chapter 8

Class Documentation

8.1 src.py.code_gen.CodeCount Class Reference

Inheritance diagram for src.py.code_gen.CodeCount:



8.1.1 Detailed Description

PoD class that contains a list of C++ lines of code and a count of the number of declarations or definitions.

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/[code_gen.py](#)

8.2 CppCustomObject Class Reference

```
#include <cUserDefined.h>
```

Public Member Functions

- [CppCustomObject](#) ()
- [CppCustomObject](#) (std::string [first](#), std::string [last](#), long [number](#))
- const std::string & [first](#) () const
- const std::string & [last](#) () const
- long [number](#) () const
- std::string [name](#) ()

8.2.1 Constructor & Destructor Documentation

8.2.1.1 CppCustomObject() [1/2]

```
CppCustomObject::CppCustomObject ( ) [inline]
```

8.2.1.2 CppCustomObject() [2/2]

```
CppCustomObject::CppCustomObject (
    std::string first,
    std::string last,
    long number ) [inline]
```

8.2.2 Member Function Documentation

8.2.2.1 first()

```
const std::string& CppCustomObject::first ( ) const [inline]
```

8.2.2.2 last()

```
const std::string& CppCustomObject::last ( ) const [inline]
```

8.2.2.3 name()

```
std::string CppCustomObject::name ( ) [inline]
```

8.2.2.4 number()

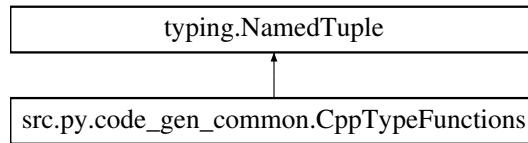
```
long CppCustomObject::number ( ) const [inline]
```

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/[cUserDefined.h](#)

8.3 src.py.code_gen_common.CppTypeFunctions Class Reference

Inheritance diagram for src.py.code_gen_common.CppTypeFunctions:



8.3.1 Detailed Description

PoD Class to contain the names of three C/C++ functions:

- Conversion from C++ to Python object. Example 'cpp_bool_to_py_bool'.
- Check it is a Python object of type. Example 'py_bool_check'.
- Conversion from Python object to a C++ type. Example 'py_bool_to_cpp_bool'.

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/[code_gen_common.py](#)

8.4 CustomObject Struct Reference

Public Attributes

- PyObject_HEAD PyObject * [first](#)
- PyObject * [last](#)
- int [number](#)

8.4.1 Member Data Documentation

8.4.1.1 first

```
PyObject_HEAD PyObject* CustomObject::first
```

8.4.1.2 last

```
PyObject* CustomObject::last
```

8.4.1.3 number

```
int CustomObject::number
```

The documentation for this struct was generated from the following file:

- [/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.cpp](#)

8.5 ExecClock Class Reference

```
#include <TestFramework.h>
```

Public Types

- `typedef std::chrono::duration< double > tHiResDouble`

Public Member Functions

- [ExecClock](#) ()
- double [seconds](#) ()

8.5.1 Member Typedef Documentation

8.5.1.1 tHiResDouble

```
typedef std::chrono::duration<double> ExecClock::tHiResDouble
```

8.5.2 Constructor & Destructor Documentation

8.5.2.1 ExecClock()

```
ExecClock::ExecClock ( ) [inline]
```

8.5.3 Member Function Documentation

8.5.3.1 seconds()

```
double ExecClock::seconds ( ) [inline]
```

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.h

8.6 std::hash< std::complex< double > > Struct Reference

```
#include <python_convert.h>
```

Public Member Functions

- `size_t operator()` (std::complex< double > const &item) const

8.6.1 Detailed Description

Provide a hash function for std::complex<double>. This mimics the Python hash of complex, see `complex_hash()` typically at <https://github.com/python/cpython/blob/main/Objects/complexobject.c#L407> See also: Include/pyhash.h

8.6.2 Member Function Documentation

8.6.2.1 operator()()

```
size_t std::hash< std::complex< double > >::operator() (
    std::complex< double > const & item ) const [inline]
```

The documentation for this struct was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert.h

8.7 std::hash< std::vector< char > > Struct Reference

```
#include <python_convert.h>
```

Public Member Functions

- `size_t operator()` (std::vector< char > const &item) const

8.7.1 Detailed Description

Provide a hash function for `std::vector<char>`. This just creates a `std::string_view` over the raw data and uses `std::hash` on that.

8.7.2 Member Function Documentation

8.7.2.1 `operator()`

```
size_t std::hash< std::vector< char > >::operator() (
    std::vector< char > const & item ) const [inline]
```

The documentation for this struct was generated from the following file:

- [/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert.h](#)

8.8 `std::less< std::complex< T > >` Struct Template Reference

```
#include <python_convert.h>
```

Public Member Functions

- `bool operator()` (`std::complex< T > const &a`, `std::complex< T > const &b`) `const`

8.8.1 Member Function Documentation

8.8.1.1 `operator()`

```
template<typename T >
bool std::less< std::complex< T > >::operator() (
    std::complex< T > const & a,
    std::complex< T > const & b ) const [inline]
```

The documentation for this struct was generated from the following file:

- [/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert.h](#)

8.9 RSSSnapshot Class Reference

```
#include <get_rss.h>
```


Public Member Functions

- [RSSSnapshot](#) (const std::string [name](#))
- const std::string & [name](#) () const
- double [rss_initial_mb](#) () const
- double [rss_peak_initial_mb](#) () const
- double [rss_now_mb](#) () const
- double [rss_peak_now_mb](#) () const
- double [rss_now_diff_mb](#) () const
- double [rss_peak_diff_mb](#) () const
- size_t [rss_initial_pages](#) () const
- size_t [rss_peak_initial_pages](#) () const
- size_t [rss_now_pages](#) () const
- size_t [rss_peak_now_pages](#) () const
- long [rss_now_diff_pages](#) () const
- long [rss_peak_diff_pages](#) () const

Protected Attributes

- std::string [m_name](#)
- size_t [m_rss_initial](#)
- size_t [m_rss_peak_initial](#)

8.9.1 Detailed Description

Class that takes a snapshot of RSS usage.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 RSSSnapshot()

```
RSSSnapshot::RSSSnapshot (  
    const std::string name ) [inline]
```

8.9.3 Member Function Documentation

8.9.3.1 name()

```
const std::string& RSSSnapshot::name ( ) const [inline]
```

8.9.3.2 rss_initial_mb()

```
double RSSSnapshot::rss_initial_mb ( ) const [inline]
```

8.9.3.3 rss_initial_pages()

```
size_t RSSSnapshot::rss_initial_pages ( ) const [inline]
```

8.9.3.4 rss_now_diff_mb()

```
double RSSSnapshot::rss_now_diff_mb ( ) const [inline]
```

8.9.3.5 rss_now_diff_pages()

```
long RSSSnapshot::rss_now_diff_pages ( ) const [inline]
```

8.9.3.6 rss_now_mb()

```
double RSSSnapshot::rss_now_mb ( ) const [inline]
```

8.9.3.7 rss_now_pages()

```
size_t RSSSnapshot::rss_now_pages ( ) const [inline]
```

8.9.3.8 rss_peak_diff_mb()

```
double RSSSnapshot::rss_peak_diff_mb ( ) const [inline]
```

8.9.3.9 rss_peak_diff_pages()

```
long RSSSnapshot::rss_peak_diff_pages ( ) const [inline]
```

8.9.3.10 rss_peak_initial_mb()

```
double RSSSnapshot::rss_peak_initial_mb ( ) const [inline]
```

8.9.3.11 rss_peak_initial_pages()

```
size_t RSSSnapshot::rss_peak_initial_pages ( ) const [inline]
```

8.9.3.12 rss_peak_now_mb()

```
double RSSSnapshot::rss_peak_now_mb ( ) const [inline]
```

8.9.3.13 rss_peak_now_pages()

```
size_t RSSSnapshot::rss_peak_now_pages ( ) const [inline]
```

8.9.4 Member Data Documentation

8.9.4.1 m_name

```
std::string RSSSnapshot::m_name [protected]
```

8.9.4.2 m_rss_initial

```
size_t RSSSnapshot::m_rss_initial [protected]
```

8.9.4.3 m_rss_peak_initial

```
size_t RSSSnapshot::m_rss_peak_initial [protected]
```

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/[get_rss.h](#)

8.10 StreamFormatState Class Reference

```
#include <save_stream_state.h>
```

Public Member Functions

- [StreamFormatState](#) (std::basic_ios< char > &stream)
- [~StreamFormatState](#) ()

8.10.1 Detailed Description

A class that saves the formatting state of a stream and restores it on destruction.

Usage:

```
{  
    StreamFormatState stream_state(os); // Stream state captured.  
    // ...  
} // Stream state restored.
```

The state that is saved is the .flags() and the .fill() character.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 StreamFormatState()

```
StreamFormatState::StreamFormatState (  
    std::basic_ios< char > & stream ) [inline]
```

Takes a stream and records its formatting state.

Parameters

<i>stream</i>	The stream.
---------------	-------------

8.10.2.2 ~StreamFormatState()

```
StreamFormatState::~StreamFormatState ( ) [inline]
```

Destruction. This restores the formatting state of the stream.

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/[save_stream_state.h](#)

8.11 SubTestCount Class Reference

```
#include <TestFramework.h>
```

Public Member Functions

- [SubTestCount](#) ()
- int [failure](#) () const
- size_t [test_count](#) () const
- void [test](#) (bool condition)
- std::vector< size_t > [test_failures](#) () const

Protected Attributes

- int [m_failure](#)
- size_t [m_test_count](#)

8.11.1 Detailed Description

In a single test function there might be many tests, this class keeps track of which sub-test is being evaluated and uniquely sets the failure flag by bit twiddling.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 SubTestCount()

```
SubTestCount::SubTestCount ( ) [inline]
```

8.11.3 Member Function Documentation

8.11.3.1 failure()

```
int SubTestCount::failure ( ) const [inline]
```

8.11.3.2 test()

```
void SubTestCount::test (
    bool condition )
```

8.11.3.3 test_count()

```
size_t SubTestCount::test_count ( ) const [inline]
```

8.11.3.4 test_failures()

```
std::vector< size_t > SubTestCount::test_failures ( ) const
```

8.11.4 Member Data Documentation

8.11.4.1 m_failure

```
int SubTestCount::m_failure [protected]
```

8.11.4.2 m_test_count

```
size_t SubTestCount::m_test_count [protected]
```

The documentation for this class was generated from the following files:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/[TestFramework.h](#)
- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/[TestFramework.cpp](#)

8.12 TestResult Class Reference

```
#include <TestFramework.h>
```

Public Member Functions

- `TestResult` (std::string `name`, int `failed`, double `execTime`, size_t `testCount`=1, size_t `scale`=1)
- `TestResult` (std::string `name`)
- void `setFailed` (size_t `scale`=1)

Setters:

- void `execTimeAdd` (int `failed`, double `execTime`, size_t `test_count`, size_t `scale`=1)
- std::string `name` () const

Getters:

- int `failed` () const
- int `failed` (size_t `scale`) const
- size_t `numTests` () const
- size_t `numTests` (size_t `scale`) const
- size_t `testCount` (size_t `scale`) const
- double `totalTime` (size_t `scale`) const
- double `execTime` (size_t `scale`) const
- double `atomicTestMeanExecTime` (size_t `scale`) const
- double `execTimeStdDev` (size_t `scale`) const
- bool `hasExecTimeStdDev` (size_t `scale`) const
- double `execTimeMin` (size_t `scale`) const
- double `execTimeMax` (size_t `scale`) const
- size_t `numScaleValues` () const
- std::vector< size_t > `scaleValues` () const
- `TestResult` (const `TestResult` &`rhs`)=default
- `TestResult` & `operator=` (const `TestResult` &`rhs`)=default
- `TestResult` (`TestResult` &&)=default

8.12.1 Detailed Description

`TestResult` is a class that retains correctness and performance information of a particular test.

Terminology: `scale` - The declared scale of the test for time complexity analysis. `repeat` - The number of times an identical test is conducted to get a statistical feel for the execution time. `count` - The number of atomic operations that the execution time has measured.

NOTE: `execTime()` gives the mean time of all of the executions. This needs to be divided by `count` to get the execution time of a single operation.

For example timing sorting a vector. `scale`: might have a range 32, 64, 128, 256, 512, 1024 (vector length). `repeat` might be 10 to get some min/max/mean/std. dev. values for variety. `count`: might be 100 to get a reasonable (single) execution time for each test.

Typical usage: `test_something()` does a single test, possibly some fast operation count number of times to get a decent time.

`test_multiple()` does the same but repeats it 10 times to get some statistics on it.

`test_complexity()` does the same as `test_multiple()` but over a range of values of `N` so that some sense of the time complexity can be gained.

```
void test_something(TestResults &results) {
    ExecClock clk;
    // Do something count number of times here, set failure to 0 | 1.
    results.push_back(TestResult(__FUNCTION__, failure, clk.execTime(), count));
}

void test_multiple(TestResults &results) {
    TestResult test_result(__FUNCTION__);
    for (size_t i = 0; i < 10; ++i) {
        ExecClock clk;
```

```

        for (size_t j = 0; j < COUNT; ++j) {
            // Do something small here COUNT number of times
        }
        test_result.execTimeAdd(0, clk.execTime(), COUNT);
    }
    results.push_back(test_result);
}

void test_complexity(TestResults &results) {
    int failure = 0;
    TestResult test_result(__FUNCTION__);
    std::list<size_t> scales = {1024, 2048, 4096, 8192};
    for (auto scale: scales) {
        // Create something appropriate to scale
        vector<char> input;
        for (tTimeVectWord i = 0; i < scale; ++i) {
            input.push_back(i);
        }
        for (size_t repeat = 0; repeat < 10; ++repeat) {
            ExecClock clk;
            // Do something small here COUNT number of times, set failure.
            test_result.execTimeAdd(failure, clk.execTime(), COUNT, scale);
        }
    }
    results.push_back(test_result);
}

int main() {
    TestResults results;
    test_something(results);
    test_multiple(results);
    test_complexity(results);
    std::cout << results;
}

```

8.12.1.1 Note on gnuplot

See also: `std::ostream &operator<<(std::ostream &os, const TestResult &result)`

This example is if the following tests are reported:

```

// Add 4 (mock_repeat) tests results
mock_test_result.execTimeAdd(0, 1.0, 100, 99);
mock_test_result.execTimeAdd(0, 2.0, 100, 99);
mock_test_result.execTimeAdd(0, 3.0, 100, 99);
mock_test_result.execTimeAdd(0, 4.0, 100, 99);

```

If the test output is put into individual .dat files such as:

#HEAD:	Fail	Scale	Repeat	Mean (s)	Std.Dev. (s)	Min. (s)	Max. (s)	Count
	Rate (/s)	Name						
#	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8
		\$10	\$11					\$9
TEST:	0	99	4	2.500000000	1.118033989	1.000000000	4.000000000	400
	160.0	test_internal_test_result_string						

So:

- The total time to run all of these tests is: ($\$5 * \4), in this case 10.0 (s).
- The mean time for each atomic test is: ($\$5 * \$4 / \$9$), in this case $10.0 / 400 = 0.025$ (s)
- The standard deviation atomic time is: ($\$6 / (\$9 / \$4)$) or ($\$6 * \$4 / \9), in this case $4 * 1.11 / 400 = +/- 0.0111$ (s)
- The minimum atomic time is: ($\$7 / (\$9 / \$4)$) or ($\$7 * \$4 / \9), in this case $4 * 1.0 / 400 = 0.010$ (s)
- The maximum atomic time is: ($\$8 / (\$9 / \$4)$) or ($\$8 * \$4 / \9), in this case $4 * 4.0 / 400 = 0.040$ (s)

Candlestick plots, from gnuplot: "Both require five columns of data: the x value, followed (in order) by the opening, low, high, and closing prices."

So a suitable gnuplot statement for a candle stick plot with time in μ s is:

```
opening : Mean - Std.Dev   (($5 - $6) * $4 / $9)
low      : Min             ($7 * $4 / $9)
high     : Max             ($8 * $4 / $9)
closing  : Mean + Std.Dev   (($5 + $6) * $4 / $9)
```

And for a suitable gnuplot statement for a candle stick plot with rate in μ s / object is:

```
opening : Mean - Std.Dev   (($5 - $6) * $4 / ($9 * $3))
low      : Min             ($7 * $4 / ($9 * $3))
high     : Max             ($8 * $4 / ($9 * $3))
closing  : Mean + Std.Dev   (($5 + $6) * $4 / ($9 * $3))
```

So for a time plot:

```
plot "dat/test_cpp_vector_char_to_py_bytes.dat" \
    using 3:($5 * ($5 - $6) * $4 / $9):($5 * $7 * $4 / $9):($5 * $8 * $4 / $9):($5 * ($5 + $6) * $4 / $9) \
    t "Python str 8 bit -> C++" with candlesticks whiskerbars 0.5 linewidth 2, \
    "dat/test_cpp_vector_char_to_py_bytes.dat" using 3:($5 * $7 * $4 / $9) \
    t "Minimum Python str 8 bit -> C++" with lines linewidth 2, \
```

And for a rate plot:

```
plot "dat/test_cpp_vector_char_to_py_bytes.dat" \
    using 3:($5 * ($5 - $6) * $4 / ($9 * $3)):($5 * $7 * $4 / ($9 * $3)):($5 * $8 * $4 / ($9 * $3)):($5 * ($5 + $6) * $4 / ($9 * $3)) \
    t "Python str 8 bit -> C++" with candlesticks whiskerbars 0.5 linewidth 2, \
    "dat/test_cpp_vector_char_to_py_bytes.dat" using 3:($5 * $7 * $4 / ($9 * $3)) \
    t "Minimum Python str 8 bit -> C++" with lines linewidth 2, \
```

NOTE: Time plot was, in error:

```
plot "dat/test_cpp_vector_char_to_py_bytes.dat" \
    using 3:($5 * ($5 - $6) / $4):($5 * $7 / $4):($5 * $8 / $4):($5 * ($5 + $6) / $4) \
    t "Python str 8 bit -> C++" with candlesticks whiskerbars 0.5 linewidth 2, \
    "dat/test_cpp_vector_char_to_py_bytes.dat" using 3:($7 * $5 / $4) \
    t "Minimum Python str 8 bit -> C++" with lines linewidth 2, \
```

8.12.1.2 Note on Using the Rate Column

If the mean value is very small it will loose resolution. For example here is a test that takes 1.23456789 ns that only has 1 S.F. resolution. Inverting the rate can recover the time to a much higher resolution.

HEAD:	Fail	Scale	Repeat	Mean(s)	Std.Dev.(s)	Min.(s)	Max.(s)	Count
		Rate(/s)	Name					
TEST:	0	99	1	0.000000001	N/A	N/A	N/A	1
		810000007.4	test_internal_test_result_string_using_rate					

8.12.2 Constructor & Destructor Documentation

8.12.2.1 TestResult() [1/4]

```
TestResult::TestResult (
    std::string name,
    int failed,
    double execTime,
    size_t testCount = 1,
    size_t scale = 1 )
```

8.12.2.2 `TestResult()` [2/4]

```
TestResult::TestResult (
    std::string name ) [inline]
```

8.12.2.3 `TestResult()` [3/4]

```
TestResult::TestResult (
    const TestResult & rhs ) [default]
```

8.12.2.4 `TestResult()` [4/4]

```
TestResult::TestResult (
    TestResult && ) [default]
```

8.12.3 Member Function Documentation

8.12.3.1 `atomicTestMeanExecTime()`

```
double TestResult::atomicTestMeanExecTime (
    size_t scale ) const
```

Returns the sum of the execution times divided by the number of atomic tests.

8.12.3.2 `execTime()`

```
double TestResult::execTime (
    size_t scale ) const
```

Returns the mean of the execution times. This does not take into account the atomic test counts.

8.12.3.3 `execTimeAdd()`

```
void TestResult::execTimeAdd (
    int failed,
    double execTime,
    size_t test_count,
    size_t scale = 1 )
```

8.12.3.4 execTimeMax()

```
double TestResult::execTimeMax (
    size_t scale ) const
```

8.12.3.5 execTimeMin()

```
double TestResult::execTimeMin (
    size_t scale ) const
```

8.12.3.6 execTimeStdDev()

```
double TestResult::execTimeStdDev (
    size_t scale ) const
```

8.12.3.7 failed() [1/2]

```
int TestResult::failed ( ) const
```

8.12.3.8 failed() [2/2]

```
int TestResult::failed (
    size_t scale ) const
```

8.12.3.9 hasExecTimeStdDev()

```
bool TestResult::hasExecTimeStdDev (
    size_t scale ) const [inline]
```

8.12.3.10 name()

```
std::string TestResult::name ( ) const [inline]
```

Getters.

8.12.3.11 numScaleValues()

```
size_t TestResult::numScaleValues ( ) const
```

8.12.3.12 numTests() [1/2]

```
size_t TestResult::numTests ( ) const
```

8.12.3.13 numTests() [2/2]

```
size_t TestResult::numTests (
    size_t scale ) const
```

8.12.3.14 operator=()

```
TestResult& TestResult::operator= (
    const TestResult & rhs ) [default]
```

8.12.3.15 scaleValues()

```
std::vector< size_t > TestResult::scaleValues ( ) const
```

8.12.3.16 setFailed()

```
void TestResult::setFailed (
    size_t scale = 1 ) [inline]
```

Setters:

8.12.3.17 testCount()

```
size_t TestResult::testCount (
    size_t scale ) const [inline]
```

8.12.3.18 totalTime()

```
double TestResult::totalTime (
    size_t scale ) const
```

The documentation for this class was generated from the following files:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/[TestFramework.h](#)
- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/[TestFramework.cpp](#)

8.13 TestResultS Class Reference

```
#include <TestFramework.h>
```

Public Types

- typedef std::list< [TestResult](#) > [tResults](#)

Public Member Functions

- [TestResultS](#) ()
- void [push_back](#) (const [TestResult](#) &result)
- const [tResults](#) & [results](#) () const
- void [dump_header](#) (std::ostream &os) const
- void [dump_tests](#) (std::ostream &os) const
- void [dump_tail](#) (std::ostream &os) const
- int [failed](#) () const

8.13.1 Detailed Description

This just accumulates a list of [TestResult](#) objects and can print them out in a human readable form.

8.13.2 Member Typedef Documentation

8.13.2.1 tResults

```
typedef std::list<TestResult> TestResultS::tResults
```

8.13.3 Constructor & Destructor Documentation

8.13.3.1 TestResultS()

```
TestResultS::TestResultS ( ) [inline]
```

8.13.4 Member Function Documentation

8.13.4.1 dump_header()

```
void TestResultS::dump_header (
    std::ostream & os ) const
```

8.13.4.2 dump_tail()

```
void TestResultS::dump_tail (
    std::ostream & os ) const
```

8.13.4.3 dump_tests()

```
void TestResultS::dump_tests (
    std::ostream & os ) const
```

8.13.4.4 failed()

```
int TestResultS::failed ( ) const [inline]
```

8.13.4.5 push_back()

```
void TestResultS::push_back (
    const TestResult & result )
```

8.13.4.6 results()

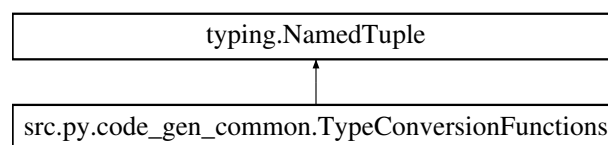
```
const tResults& TestResultS::results ( ) const [inline]
```

The documentation for this class was generated from the following files:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.h
- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.cpp

8.14 src.py.code_gen_common.TypeConversionFunctions Class Reference

Inheritance diagram for src.py.code_gen_common.TypeConversionFunctions:



8.14.1 Detailed Description

PoD Class to contain the names of three C/C++ functions:

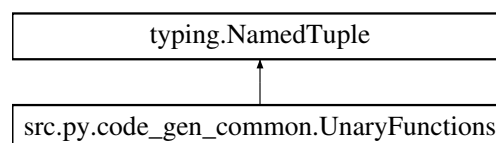
- Check it is a Python object of type. Example 'py_bool_check'.
- Conversion from Python object to a C++ type. Example 'py_bool_to_cpp_bool'.
- Conversion from C++ to Python object. Example 'cpp_bool_to_py_bool'.

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_common.py

8.15 src.py.code_gen_common.UnaryFunctions Class Reference

Inheritance diagram for src.py.code_gen_common.UnaryFunctions:



8.15.1 Detailed Description

PoD Class to contain the names of three C/C++ functions:

- Python container type. Example 'list'.
- C++ container type. Example 'std::vector'.
- Function declaration to convert to a Python type. Example 'cpp_std_list_like_to_py_tuple'.
- Function declaration to convert to a C++ type. Example 'py_tuple_to_cpp_std_list_like'.

The documentation for this class was generated from the following file:

- /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_common.py

Chapter 9

File Documentation

9.1 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/___init___.py File Reference

Namespaces

- [src](#)

9.2 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/___init___.py File Reference

Namespaces

- [src.py](#)

9.3 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/get_rss.cpp File Reference

```
#include <iomanip>
#include "get_rss.h"
#include "save_stream_state.h"
```

Macros

- `#define RSS_SNAPSHOT_REPORT_PAGES 0`

Functions

- `size_t` [getPeakRSS](#) (void)
- `size_t` [getCurrentRSS](#) (void)
- `size_t` [getCurrentRSS_alterate](#) (void)
- `double` [getPeakRSSMb](#) ()
- `double` [getCurrentRSSMb](#) ()
- `double` [getCurrentRSS_alterateMb](#) ()
- `std::ostream &` [operator<<](#) (`std::ostream &os`, `const RSSSnapshot &rss`)

Variables

- `const double` [MEGABYTES](#) = (1 << 20)
- `static const int` [MB_PRECISION](#) = 3
- `static const int` [MB_WIDTH](#) = 10

9.3.1 Macro Definition Documentation

9.3.1.1 RSS_SNAPSHOT_REPORT_PAGES

```
#define RSS_SNAPSHOT_REPORT_PAGES 0
```

9.3.2 Function Documentation

9.3.2.1 getCurrentRSS()

```
size_t getCurrentRSS (
    void )
```

Returns the current resident set size (physical memory use) measured in bytes, or zero if the value cannot be determined on this OS.

9.3.2.2 getCurrentRSS_alterate()

```
size_t getCurrentRSS_alterate ( )
```

Return the current RSS in bytes.

Returns

9.3.2.3 `getCurrentRSS_alterateMb()`

```
double getCurrentRSS_alterateMb ( )
```

9.3.2.4 `getCurrentRSSMb()`

```
double getCurrentRSSMb ( )
```

9.3.2.5 `getPeakRSS()`

```
size_t getPeakRSS (
    void )
```

Returns the peak (maximum so far) resident set size (physical memory use) measured in bytes, or zero if the value cannot be determined on this OS.

9.3.2.6 `getPeakRSSMb()`

```
double getPeakRSSMb ( )
```

9.3.2.7 `operator<<()`

```
std::ostream& operator<< (
    std::ostream & os,
    const RSSSnapshot & rss )
```

9.3.3 Variable Documentation

9.3.3.1 `MB_PRECISION`

```
const int MB_PRECISION = 3 [static]
```

9.3.3.2 `MB_WIDTH`

```
const int MB_WIDTH = 10 [static]
```

9.3.3.3 MEGABYTES

```
const double MEGABYTES = (1 << 20)
```

Size of one megabyte in bytes, $1 \ll 20$

9.4 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/get_rss.h File Reference

```
#include <iostream>
#include <stdlib.h>
```

Classes

- class [RSSSnapshot](#)

Functions

- `size_t` [getPeakRSS](#) ()
- `size_t` [getCurrentRSS](#) ()
- `size_t` [getCurrentRSS_alternate](#) ()
- `std::ostream &` [operator<<](#) (`std::ostream &os`, `const RSSSnapshot &rss`)

Variables

- `const double` [MEGABYTES](#)

9.4.1 Function Documentation

9.4.1.1 `getCurrentRSS()`

```
size_t getCurrentRSS (
    void )
```

Return the current RSS in bytes.

Returns

Returns the current resident set size (physical memory use) measured in bytes, or zero if the value cannot be determined on this OS.

9.4.1.2 getCurrentRSS_alterate()

```
size_t getCurrentRSS_alterate ( )
```

Return the current RSS in bytes.

Returns

9.4.1.3 getPeakRSS()

```
size_t getPeakRSS (
    void )
```

Return the peak RSS in bytes.

Returns

Returns the peak (maximum so far) resident set size (physical memory use) measured in bytes, or zero if the value cannot be determined on this OS.

9.4.1.4 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const RSSSnapshot & rss )
```

9.4.2 Variable Documentation

9.4.2.1 MEGABYTES

```
const double MEGABYTES [extern]
```

Size of one megabyte in bytes, 1 << 20

9.5 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/save_stream_state.h File Reference

```
#include <iostream>
#include <iomanip>
```

Classes

- class [StreamFormatState](#)

9.6 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.cpp File Reference

```
#include <iomanip>
#include <cmath>
#include <cassert>
#include <sstream>
#include "TestFramework.h"
#include "save_stream_state.h"
```

Macros

- #define [REGEX_SPACE_INTEGER](#) "\\s+(\\d+)"
- #define [REGEX_SPACE_FLOAT](#) "\\s+([0-9+-.]+)"
- #define [REGEX_SPACE_STRING_NO_SPACE](#) "\\s+(\\S+)"
- #define [REGEX_SPACE_ANYTHING](#) "\\s+(.+)"

Functions

- std::ostream & [operator<<](#) (std::ostream &os, const [TestResult](#) &result)
- std::ostream & [operator<<](#) (std::ostream &os, const [TestResultS](#) &results)
- std::string [unique_string](#) (int width)
- std::u16string [unique_u16string](#) (int width)
- std::u32string [unique_u32string](#) (int width)
- std::vector< char > [unique_vector_char](#) (int width)
- size_t [count_of_unique_string](#) ()
- void [reset_count_of_unique_string](#) ()

Variables

- static const int [TIME_PRECISION](#) = 9
- static const int [TIME_WIDTH](#) = 16
- static size_t [str_count](#) = 0

9.6.1 Macro Definition Documentation

9.6.1.1 REGEX_SPACE_ANYTHING

```
#define REGEX_SPACE_ANYTHING "\\s+(.+)"
```

9.6.1.2 REGEX_SPACE_FLOAT

```
#define REGEX_SPACE_FLOAT "\\s+([0-9+-.]+)"
```

9.6.1.3 REGEX_SPACE_INTEGER

```
#define REGEX_SPACE_INTEGER "\\s+(\\d+)"
```

9.6.1.4 REGEX_SPACE_STRING_NO_SPACE

```
#define REGEX_SPACE_STRING_NO_SPACE "\\s+(\\S+)"
```

9.6.2 Function Documentation

9.6.2.1 count_of_unique_string()

```
size_t count_of_unique_string ( )
```

Returns the number of unique strings created.

Returns

Number of unique strings created.

9.6.2.2 operator<<() [1/2]

```
std::ostream& operator<< (
    std::ostream & os,
    const TestResult & result )
```

9.6.2.3 Note on The Output

Example output (with header and \$ gnuplot guide).

```
#HEAD: Fail    Scale  Repeat      Mean (s)    Std.Dev. (s)      Min. (s)      Max. (s)      Count
#      $1     $2     $3     $4     $5     $6     $7     $8     $9
#      $10 $11
TEST:  0      99      4      2.500000000  1.118033989  1.000000000  4.000000000  400
      160.0 test_internal_test_result_string
```

In this case we have repeated the tests four times, each of these tests has a timed operation (the 'atomic' test) that is repeated 100 times for each individual execution time.

The total time to run all of these tests is: Repeat * Mean in seconds, in this case 10.0 (s). The mean time for each atomic test is: Repeat * Mean / Count, in this case $10.0 / 400 = 0.025$ (s) The minimum atomic time is: Repeat * Min.(s) / Count, in this case $4 * 1.0 / 400 = 0.010$ (s) The maximum atomic time is: Repeat * Max.(s) / Count, in this case $4 * 4.0 / 400 = 0.040$ (s) The standard deviation atomic time is: Repeat * Std.Dev.(s) / Count, in this case $4 * 1.11 / 400 = +/- 0.0111$ (s)

Parameters

<i>os</i>	The stream to write to.
<i>result</i>	The tests result.

Returns

The stream for chaining.

9.6.2.4 operator<<() [2/2]

```
std::ostream& operator<< (
    std::ostream & os,
    const TestResults & results )
```

9.6.2.5 reset_count_of_unique_string()

```
void reset_count_of_unique_string ( )
```

Reset the string counter.

9.6.2.6 unique_string()

```
std::string unique_string (
    int width )
```

Creates a unique `std::string`. This starts as "0", "1" ...

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::string`.

9.6.2.7 unique_u16string()

```
std::u16string unique_u16string (
    int width )
```

Creates a unique `std::u16string`. This starts as "0", "1" ...

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::u16string`.

9.6.2.8 unique_u32string()

```
std::u32string unique_u32string (  
    int width )
```

Creates a unique `std::u32string`. This starts as "0", "1" ...

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::u32string`.

9.6.2.9 unique_vector_char()

```
std::vector<char> unique_vector_char (  
    int width )
```

Creates a unique `std::vector<char>`.

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::vector<char>`.

9.6.3 Variable Documentation

9.6.3.1 str_count

```
size_t str_count = 0 [static]
```

9.6.3.2 TIME_PRECISION

```
const int TIME_PRECISION = 9 [static]
```

9.6.3.3 TIME_WIDTH

```
const int TIME_WIDTH = 16 [static]
```

9.7 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpp/TestFramework.h File Reference

```
#include <list>
#include <map>
#include <string>
#include <vector>
#include <iostream>
#include <chrono>
```

Classes

- class [ExecClock](#)
- class [TestResult](#)
- class [TestResults](#)
- class [SubTestCount](#)

Functions

- std::ostream & [operator<<](#) (std::ostream &os, const [TestResult](#) &result)
- std::ostream & [operator<<](#) (std::ostream &os, const [TestResults](#) &results)
- std::string [unique_string](#) (int width=0)
- std::u16string [unique_u16string](#) (int width=0)
- std::u32string [unique_u32string](#) (int width=0)
- std::vector< char > [unique_vector_char](#) (int width)
- size_t [count_of_unique_string](#) ()
- void [reset_count_of_unique_string](#) ()

9.7.1 Function Documentation

9.7.1.1 `count_of_unique_string()`

```
size_t count_of_unique_string ( )
```

Returns the number of unique strings created.

Returns

Number of unique strings created.

9.7.1.2 `operator<<()` [1/2]

```
std::ostream& operator<< (
    std::ostream & os,
    const TestResult & result )
```

9.7.1.3 Note on The Output

Example output (with header and \$ gnuplot guide).

```
#HEAD: Fail    Scale  Repeat      Mean (s)      Std.Dev. (s)      Min. (s)      Max. (s)      Count
      Rate (/s) Name
#  $1  $2    $3      $4          $5          $6          $7          $8          $9
      $10 $11
TEST:  0      99      4      2.500000000    1.118033989    1.000000000    4.000000000    400
      160.0 test_internal_test_result_string
```

In this case we have repeated the tests four times, each of these tests has a timed operation (the 'atomic' test) that is repeated 100 times for each individual execution time.

The total time to run all of these tests is: Repeat * Mean in seconds, in this case 10.0 (s). The mean time for each atomic test is: Repeat * Mean / Count, in this case 10.0 / 400 = 0.025 (s) The minimum atomic time is: Repeat * Min.(s) / Count, in this case 4 * 1.0 / 400 = 0.010 (s) The maximum atomic time is: Repeat * Max.(s) / Count, in this case 4 * 4.0 / 400 = 0.040 (s) The standard deviation atomic time is: Repeat * Std.Dev.(s) / Count, in this case 4 * 1.11 / 400 = +/- 0.0111 (s)

Parameters

<i>os</i>	The stream to write to.
<i>result</i>	The tests result.

Returns

The stream for chaining.

9.7.1.4 operator<<() [2/2]

```
std::ostream& operator<< (
    std::ostream & os,
    const TestResults & results )
```

9.7.1.5 reset_count_of_unique_string()

```
void reset_count_of_unique_string ( )
```

Reset the string counter.

9.7.1.6 unique_string()

```
std::string unique_string (
    int width )
```

Creates a unique `std::string`. This starts as "0", "1" ...

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::string`.

9.7.1.7 unique_u16string()

```
std::u16string unique_u16string (
    int width )
```

Creates a unique `std::u16string`. This starts as "0", "1" ...

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::u16string`.

9.7.1.8 unique_u32string()

```
std::u32string unique_u32string (
    int width )
```

Creates a unique `std::u32string`. This starts as "0", "1" ...

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::u32string`.

9.7.1.9 unique_vector_char()

```
std::vector<char> unique_vector_char (
    int width )
```

Creates a unique `std::vector<char>`.

Parameters

<i>width</i>	If > 0 the string will be at least this width.
--------------	--

Returns

The unique `std::vector<char>`.

9.8 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/auto_py_convert_internal.cpp File Reference

```
#include "python_convert.h"
```

Namespaces

- [Python_Cpp_Containers](#)

Conversion functions for individual Python objects.

Functions

- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< bool > (const std::vector< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (const std::vector< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< double > (const std::vector< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::complex< double > > (const std::vector< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::vector< char > > (const std::vector< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::string > (const std::vector< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u16string > (const std::vector< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u32string > (const std::vector< std::u32string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< bool > (PyObject *op, std::vector< bool > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< long > (PyObject *op, std::vector< long > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< double > (PyObject *op, std::vector< double > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::vector< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::vector< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::string > (PyObject *op, std::vector< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u16string > (PyObject *op, std::vector< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u32string > (PyObject *op, std::vector< std::u32string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< bool > (const std::list< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (const std::list< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< double > (const std::list< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::complex< double > > (const std::list< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::vector< char > > (const std::list< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::string > (const std::list< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u16string > (const std::list< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u32string > (const std::list< std::u32string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< bool > (PyObject *op, std::list< bool > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< double > (PyObject *op, std::list< double > &container)`

- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::list< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::list< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::string > (PyObject *op, std::list< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u16string > (PyObject *op, std::list< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u32string > (PyObject *op, std::list< std::u32string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< bool > (const std::vector< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (const std::vector< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< double > (const std::vector< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::complex< double > > (const std::vector< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::vector< char > > (const std::vector< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::string > (const std::vector< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u16string > (const std::vector< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u32string > (const std::vector< std::u32string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< bool > (PyObject *op, std::vector< bool > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (PyObject *op, std::vector< long > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< double > (PyObject *op, std::vector< double > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::vector< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::vector< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::string > (PyObject *op, std::vector< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u16string > (PyObject *op, std::vector< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u32string > (PyObject *op, std::vector< std::u32string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< bool > (const std::list< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (const std::list< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< double > (const std::list< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::complex< double > > (const std::list< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::vector< char > > (const std::list< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::string > (const std::list< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u16string > (const std::list< std::u16string > &container)`

- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u32string > (const std::list< std::u32string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< bool > (PyObject *op, std::list< bool > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< double > (PyObject *op, std::list< double > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::list< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::list< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::string > (PyObject *op, std::list< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u16string > (PyObject *op, std::list< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u32string > (PyObject *op, std::list< std::u32string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< bool > (const std::unordered_set< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< long > (const std::unordered_set< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< double > (const std::unordered_set< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::complex< double > > (const std::unordered_set< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::vector< char > > (const std::unordered_set< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::string > (const std::unordered_set< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::u16string > (const std::unordered_set< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::u32string > (const std::unordered_set< std::u32string > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< bool > (PyObject *op, std::unordered_set< bool > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< long > (PyObject *op, std::unordered_set< long > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< double > (PyObject *op, std::unordered_set< double > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::complex< double > > (PyObject *op, std::unordered_set< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::vector< char > > (PyObject *op, std::unordered_set< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::string > (PyObject *op, std::unordered_set< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::u16string > (PyObject *op, std::unordered_set< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::u32string > (PyObject *op, std::unordered_set< std::u32string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< bool > (const std::unordered_set< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< long > (const std::unordered_set< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< double > (const std::unordered_set< double > &container)`

- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::complex< double > > (const std::unordered_set< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::vector< char > > (const std::unordered_set< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::string > (const std::unordered_set< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::u16string > (const std::unordered_set< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::u32string > (const std::unordered_set< std::u32string > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< bool > (PyObject *op, std::unordered_set< bool > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< long > (PyObject *op, std::unordered_set< long > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< double > (PyObject *op, std::unordered_set< double > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::complex< double > > (PyObject *op, std::unordered_set< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::vector< char > > (PyObject *op, std::unordered_set< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::string > (PyObject *op, std::unordered_set< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::u16string > (PyObject *op, std::unordered_set< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::u32string > (PyObject *op, std::unordered_set< std::u32string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool > (const std::unordered_map< bool, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, bool > (PyObject *op, std::unordered_map< bool, bool > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, long > (const std::unordered_map< bool, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, long > (PyObject *op, std::unordered_map< bool, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, double > (const std::unordered_map< bool, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, double > (PyObject *op, std::unordered_map< bool, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::complex< double > > (const std::unordered_map< bool, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::complex< double > > (PyObject *op, std::unordered_map< bool, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > > (const std::unordered_map< bool, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::vector< char > > (PyObject *op, std::unordered_map< bool, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::string > (const std::unordered_map< bool, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::string > (PyObject *op, std::unordered_map< bool, std::string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string > (const std::unordered_map< bool, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u16string > (PyObject *op, std::unordered_map< bool, std::u16string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string > (const std::unordered_map< bool, std::u32string > &map)`

- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u32string > (PyObject *op, std::unordered_map< bool, std::u32string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, bool > (const std::unordered_map< long, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, bool > (PyObject *op, std::unordered_map< long, bool > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, long > (const std::unordered_map< long, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, long > (PyObject *op, std::unordered_map< long, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, double > (const std::unordered_map< long, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, double > (PyObject *op, std::unordered_map< long, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::complex< double > > (const std::unordered_map< long, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex< double > > (PyObject *op, std::unordered_map< long, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > > (const std::unordered_map< long, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::vector< char > > (PyObject *op, std::unordered_map< long, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string > (const std::unordered_map< long, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::string > (PyObject *op, std::unordered_map< long, std::string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string > (const std::unordered_map< long, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u16string > (PyObject *op, std::unordered_map< long, std::u16string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u32string > (const std::unordered_map< long, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u32string > (PyObject *op, std::unordered_map< long, std::u32string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double, bool > (const std::unordered_map< double, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, bool > (PyObject *op, std::unordered_map< double, bool > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double, long > (const std::unordered_map< double, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, long > (PyObject *op, std::unordered_map< double, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double, double > (const std::unordered_map< double, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, double > (PyObject *op, std::unordered_map< double, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double, std::complex< double > > (const std::unordered_map< double, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::complex< double > > (PyObject *op, std::unordered_map< double, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double, std::vector< char > > (const std::unordered_map< double, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, double, std::vector< char > > (PyObject *op, std::unordered_map< double, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, double, std::string > (const std::unordered_map< double, std::string > &map)`

- [illegible]

- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, long > (PyObject *op, std::map< long, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, double > (const std::map< long, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, double > (PyObject *op, std::map< long, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::complex< double > > (const std::map< long, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::complex< double > > (PyObject *op, std::map< long, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::vector< char > > (const std::map< long, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > > (PyObject *op, std::map< long, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::string > (const std::map< long, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::string > (PyObject *op, std::map< long, std::string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::u16string > (const std::map< long, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::u16string > (PyObject *op, std::map< long, std::u16string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, std::u32string > (const std::map< long, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, std::u32string > (PyObject *op, std::map< long, std::u32string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, bool > (const std::map< double, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, bool > (PyObject *op, std::map< double, bool > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, long > (const std::map< double, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, long > (PyObject *op, std::map< double, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, double > (const std::map< double, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, double > (PyObject *op, std::map< double, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::complex< double > > (const std::map< double, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::complex< double > > (PyObject *op, std::map< double, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::vector< char > > (const std::map< double, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > > (PyObject *op, std::map< double, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::string > (const std::map< double, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::string > (PyObject *op, std::map< double, std::string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::u16string > (const std::map< double, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::u16string > (PyObject *op, std::map< double, std::u16string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::u32string > (const std::map< double, std::u32string > &map)`

- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string > (PyObject *op, std::map< std::string, std::u32string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, bool > (const std::map< std::u16string, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, bool > (PyObject *op, std::map< std::u16string, bool > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, long > (const std::map< std::u16string, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, long > (PyObject *op, std::map< std::u16string, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, double > (const std::map< std::u16string, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, double > (PyObject *op, std::map< std::u16string, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double > > (const std::map< std::u16string, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex< double > > (PyObject *op, std::map< std::u16string, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > > (const std::map< std::u16string, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector< char > > (PyObject *op, std::map< std::u16string, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string > (const std::map< std::u16string, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string > (PyObject *op, std::map< std::u16string, std::string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string > (const std::map< std::u16string, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string > (PyObject *op, std::map< std::u16string, std::u16string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string > (const std::map< std::u16string, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string > (PyObject *op, std::map< std::u16string, std::u32string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, bool > (const std::map< std::u32string, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, bool > (PyObject *op, std::map< std::u32string, bool > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, long > (const std::map< std::u32string, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, long > (PyObject *op, std::map< std::u32string, long > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, double > (const std::map< std::u32string, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, double > (PyObject *op, std::map< std::u32string, double > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::complex< double > > (const std::map< std::u32string, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex< double > > (PyObject *op, std::map< std::u32string, std::complex< double > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::vector< char > > (const std::map< std::u32string, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector< char > > (PyObject *op, std::map< std::u32string, std::vector< char > > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::string > (const std::map< std::u32string, std::string > &map)`

- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string > (PyObject *op, std::map< std::u32string, std::string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u16string > (const std::map< std::u32string, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string > (PyObject *op, std::map< std::u32string, std::u16string > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u32string > (const std::map< std::u32string, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string > (PyObject *op, std::map< std::u32string, std::u32string > &map)`

9.9 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/auto_py_convert_internal.h File Reference

```
#include <Python.h>
```

Namespaces

- [Python_Cpp_Containers](#)
Conversion functions for individual Python objects.

Functions

- `template<typename T > PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple (const std::vector< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< bool > (const std::vector< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (const std::vector< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< double > (const std::vector< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::complex< double > > (const std::vector< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::vector< char > > (const std::vector< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::string > (const std::vector< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u16string > (const std::vector< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u32string > (const std::vector< std::u32string > &container)`
- `template<typename T > int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< bool > (PyObject *op, std::vector< bool > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< long > (PyObject *op, std::vector< long > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< double > (PyObject *op, std::vector< double > &container)`

- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::vector< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::vector< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::string > (PyObject *op, std::vector< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u16string > (PyObject *op, std::vector< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u32string > (PyObject *op, std::vector< std::u32string > &container)`
- `template<typename T > PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple (const std::list< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< bool > (const std::list< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (const std::list< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< double > (const std::list< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::complex< double > > (const std::list< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::vector< char > > (const std::list< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::string > (const std::list< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u16string > (const std::list< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< std::u32string > (const std::list< std::u32string > &container)`
- `template<typename T > int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< bool > (PyObject *op, std::list< bool > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< double > (PyObject *op, std::list< double > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::list< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::list< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::string > (PyObject *op, std::list< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u16string > (PyObject *op, std::list< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_tuple_to_cpp_std_list_like< std::u32string > (PyObject *op, std::list< std::u32string > &container)`
- `template<typename T > PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list (const std::vector< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< bool > (const std::vector< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (const std::vector< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< double > (const std::vector< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::complex< double > > (const std::vector< std::complex< double > > &container)`

- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::vector< char > > (const std::vector< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::string > (const std::vector< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u16string > (const std::vector< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u32string > (const std::vector< std::u32string > &container)`
- `template<typename T > int Python_Cpp_Containers::py_list_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< bool > (PyObject *op, std::vector< bool > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (PyObject *op, std::vector< long > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< double > (PyObject *op, std::vector< double > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::vector< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::vector< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::string > (PyObject *op, std::vector< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u16string > (PyObject *op, std::vector< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u32string > (PyObject *op, std::vector< std::u32string > &container)`
- `template<typename T > PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list (const std::list< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< bool > (const std::list< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (const std::list< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< double > (const std::list< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::complex< double > > (const std::list< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::vector< char > > (const std::list< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::string > (const std::list< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u16string > (const std::list< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< std::u32string > (const std::list< std::u32string > &container)`
- `template<typename T > int Python_Cpp_Containers::py_list_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< bool > (PyObject *op, std::list< bool > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< double > (PyObject *op, std::list< double > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::complex< double > > (PyObject *op, std::list< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::vector< char > > (PyObject *op, std::list< std::vector< char > > &container)`

- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::string > (PyObject *op, std::list< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u16string > (PyObject *op, std::list< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< std::u32string > (PyObject *op, std::list< std::u32string > &container)`
- `template<typename T > PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set (const std::unordered_set< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< bool > (const std::unordered_set< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< long > (const std::unordered_set< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< double > (const std::unordered_set< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::complex< double > > (const std::unordered_set< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::vector< char > > (const std::unordered_set< std::vector< char > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::string > (const std::unordered_set< std::string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::u16string > (const std::unordered_set< std::u16string > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_set< std::u32string > (const std::unordered_set< std::u32string > &container)`
- `template<typename T > int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< bool > (PyObject *op, std::unordered_set< bool > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< long > (PyObject *op, std::unordered_set< long > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< double > (PyObject *op, std::unordered_set< double > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::complex< double > > (PyObject *op, std::unordered_set< std::complex< double > > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::vector< char > > (PyObject *op, std::unordered_set< std::vector< char > > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::string > (PyObject *op, std::unordered_set< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::u16string > (PyObject *op, std::unordered_set< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_set_to_cpp_std_unordered_set< std::u32string > (PyObject *op, std::unordered_set< std::u32string > &container)`
- `template<typename T > PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset (const std::unordered_set< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< bool > (const std::unordered_set< bool > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< long > (const std::unordered_set< long > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< double > (const std::unordered_set< double > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::complex< double > > (const std::unordered_set< std::complex< double > > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::vector< char > > (const std::unordered_set< std::vector< char > > &container)`

- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::string >`
(const std::unordered_set< std::string > &container)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::u16string >`
(const std::unordered_set< std::u16string > &container)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_unordered_set_to_py_frozenset< std::u32string >`
(const std::unordered_set< std::u32string > &container)
- `template<typename T >`
`int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set (PyObject *op, std::unordered_set<`
`T > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< bool > (PyObject *op,`
`std::unordered_set< bool > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< long > (PyObject *op,`
`std::unordered_set< long > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< double > (PyObject`
`*op, std::unordered_set< double > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::complex< double > >`
(PyObject *op, std::unordered_set< std::complex< double > > &container)
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::vector< char > >`
(PyObject *op, std::unordered_set< std::vector< char > > &container)
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::string > (Py↵`
`Object *op, std::unordered_set< std::string > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::u16string > (Py↵`
`Object *op, std::unordered_set< std::u16string > &container)`
- `template<> int Python_Cpp_Containers::py_frozenset_to_cpp_std_unordered_set< std::u32string > (Py↵`
`Object *op, std::unordered_set< std::u32string > &container)`
- `template<template< typename ... > class Map, typename K , typename V >`
`PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict (const Map< K, V > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool >`
(const std::unordered_map< bool, bool > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, long >`
(const std::unordered_map< bool, long > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, double >`
(const std::unordered_map< bool, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::complex< double > >`
(const std::unordered_map< bool, std::complex< double > > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > >`
(const std::unordered_map< bool, std::vector< char > > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::string >`
(const std::unordered_map< bool, std::string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string >`
(const std::unordered_map< bool, std::u16string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string >`
(const std::unordered_map< bool, std::u32string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, bool >`
(const std::unordered_map< long, bool > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, long >`
(const std::unordered_map< long, long > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, double >`
(const std::unordered_map< long, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::complex< double > >`
(const std::unordered_map< long, std::complex< double > > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > >`
(const std::unordered_map< long, std::vector< char > > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string >`
(const std::unordered_map< long, std::string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string >`
(const std::unordered_map< long, std::u16string > &map)

- [illegible]

- [illegible]

- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, double >`
(const std::map< double, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::complex< double > >`
(const std::map< double, std::complex< double >> &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::vector< char > >`
(const std::map< double, std::vector< char >> &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::string >`
(const std::map< double, std::string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::u16string >`
(const std::map< double, std::u16string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, double, std::u32string >`
(const std::map< double, std::u32string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::complex< double >, bool >`
(const std::map< std::complex< double >, bool > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::complex< double >, long >`
(const std::map< std::complex< double >, long > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::complex< double >, double >`
(const std::map< std::complex< double >, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::vector< char >, bool >`
(const std::map< std::vector< char >, bool > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::vector< char >, long >`
(const std::map< std::vector< char >, long > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::vector< char >, double >`
(const std::map< std::vector< char >, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, bool >`
(const std::map< std::string, bool > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, long >`
(const std::map< std::string, long > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, double >`
(const std::map< std::string, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::complex< double > >`
(const std::map< std::string, std::complex< double >> &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::vector< char > >`
(const std::map< std::string, std::vector< char >> &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::string >`
(const std::map< std::string, std::string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::u16string >`
(const std::map< std::string, std::u16string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::string, std::u32string >`
(const std::map< std::string, std::u32string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, bool >`
(const std::map< std::u16string, bool > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, long >`
(const std::map< std::u16string, long > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, double >`
(const std::map< std::u16string, double > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double > >`
(const std::map< std::u16string, std::complex< double >> &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > >`
(const std::map< std::u16string, std::vector< char >> &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string >`
(const std::map< std::u16string, std::string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string >`
(const std::map< std::u16string, std::u16string > &map)
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string >`
(const std::map< std::u16string, std::u32string > &map)

- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > > (PyObject *op, std::map< double, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::string > (PyObject *op, std::map< double, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::u16string > (PyObject *op, std::map< double, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, double, std::u32string > (PyObject *op, std::map< double, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::complex< double >, bool > (PyObject *op, std::map< std::complex< double >, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::complex< double >, long > (PyObject *op, std::map< std::complex< double >, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::complex< double >, double > (PyObject *op, std::map< std::complex< double >, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::vector< char >, bool > (PyObject *op, std::map< std::vector< char >, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::vector< char >, long > (PyObject *op, std::map< std::vector< char >, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::vector< char >, double > (PyObject *op, std::map< std::vector< char >, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, bool > (PyObject *op, std::map< std::string, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, long > (PyObject *op, std::map< std::string, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, double > (PyObject *op, std::map< std::string, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::complex< double > > (PyObject *op, std::map< std::string, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::vector< char > > (PyObject *op, std::map< std::string, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::string > (PyObject *op, std::map< std::string, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string > (PyObject *op, std::map< std::string, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string > (PyObject *op, std::map< std::string, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, bool > (PyObject *op, std::map< std::u16string, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, long > (PyObject *op, std::map< std::u16string, long > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, double > (PyObject *op, std::map< std::u16string, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex< double > > (PyObject *op, std::map< std::u16string, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector< char > > (PyObject *op, std::map< std::u16string, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string > (PyObject *op, std::map< std::u16string, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string > (PyObject *op, std::map< std::u16string, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string > (PyObject *op, std::map< std::u16string, std::u32string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, bool > (PyObject *op, std::map< std::u32string, bool > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, long > (PyObject *op, std::map< std::u32string, long > &map)`

- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, double > (PyObject *op, std::map< std::u32string, double > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex< double > > (PyObject *op, std::map< std::u32string, std::complex< double > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector< char > > (PyObject *op, std::map< std::u32string, std::vector< char > > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string > (PyObject *op, std::map< std::u32string, std::string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string > (PyObject *op, std::map< std::u32string, std::u16string > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string > (PyObject *op, std::map< std::u32string, std::u32string > &map)`

9.10 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_container_convert.cpp File Reference

```
#include "python_container_convert.h"
```

Namespaces

- [Python_Cpp_Containers](#)
Conversion functions for individual Python objects.

Functions

- `int Python_Cpp_Containers::py_tuple_check (PyObject *op)`
- `PyObject * Python_Cpp_Containers::py_tuple_new (size_t len)`
- `Py_ssize_t Python_Cpp_Containers::py_tuple_len (PyObject *op)`
- `int Python_Cpp_Containers::py_tuple_set (PyObject *tuple_p, size_t pos, PyObject *op)`
- `PyObject * Python_Cpp_Containers::py_tuple_get (PyObject *tuple_p, size_t pos)`
- `int Python_Cpp_Containers::py_list_check (PyObject *op)`
- `PyObject * Python_Cpp_Containers::py_list_new (size_t len)`
- `Py_ssize_t Python_Cpp_Containers::py_list_len (PyObject *op)`
- `int Python_Cpp_Containers::py_list_set (PyObject *list_p, size_t pos, PyObject *op)`
- `PyObject * Python_Cpp_Containers::py_list_get (PyObject *list_p, size_t pos)`
- `int Python_Cpp_Containers::py_set_check (PyObject *op)`
- `int Python_Cpp_Containers::py_frozenset_check (PyObject *op)`

9.11 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_container_convert.h File Reference

```
#include <Python.h>
```

Namespaces

- [Python_Cpp_Containers](#)

Conversion functions for individual Python objects.

Functions

- int [Python_Cpp_Containers::py_tuple_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::py_tuple_new](#) (size_t len)
- Py_ssize_t [Python_Cpp_Containers::py_tuple_len](#) (PyObject *op)
- int [Python_Cpp_Containers::py_tuple_set](#) (PyObject *tuple_p, size_t pos, PyObject *op)
- PyObject * [Python_Cpp_Containers::py_tuple_get](#) (PyObject *tuple_p, size_t pos)
- int [Python_Cpp_Containers::py_list_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::py_list_new](#) (size_t len)
- Py_ssize_t [Python_Cpp_Containers::py_list_len](#) (PyObject *op)
- int [Python_Cpp_Containers::py_list_set](#) (PyObject *list_p, size_t pos, PyObject *op)
- PyObject * [Python_Cpp_Containers::py_list_get](#) (PyObject *list_p, size_t pos)
- int [Python_Cpp_Containers::py_set_check](#) (PyObject *op)
- int [Python_Cpp_Containers::py_frozenset_check](#) (PyObject *op)

9.12 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert.h File Reference

```
#include <Python.h>
#include <array>
#include <complex>
#include <list>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <vector>
#include "python_object_convert.h"
#include "python_container_convert.h"
#include "auto_py_convert_internal.h"
```

Classes

- struct [std::hash< std::vector< char > >](#)
- struct [std::hash< std::complex< double > >](#)
- struct [std::less< std::complex< T > >](#)

Namespaces

- [std](#)
- [Python_Cpp_Containers](#)

Conversion functions for individual Python objects.

Macros

- `#define PYTHON_CPP_CONTAINERS_VERSION "0.3.2"`

Conversion functions for individual Python objects.

Enumerations

- enum class `Python_Cpp_Containers::ErrorReturnValue` : int {
`Python_Cpp_Containers::SUCCESS` = 0 , `Python_Cpp_Containers::FAIL_CONTAINER_WRONG_TYPE` ,
`Python_Cpp_Containers::FAIL_CONTAINER_MEMBER_WRONG_TYPE` , `Python_Cpp_Containers::FAIL_CONTAINER_KEY_`
`,`
`Python_Cpp_Containers::FAIL_CONTAINER_VALUE_WRONG_TYPE` }

Functions

- `template<template< typename ... > class ListLike, typename T , PyObject (*)(const T &) ConvertCppToPy, PyObject (*)(size_t) PyUnaryContainer_New, int (*)(PyObject *, size_t, PyObject *) PyUnaryContainer_Set>`
`PyObject * Python_Cpp_Containers::very_generic_cpp_std_list_like_to_py_unary` (const ListLike< T > &list_like)
- `template<template< typename ... > class ListLike, typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert, int (*)(PyObject *) PyUnaryContainer_Check, Py_ssize_t (*)(PyObject *) PyUnaryContainer_Size, PyObject (*)(PyObject *, size_t) PyUnaryContainer_Get>`
`int Python_Cpp_Containers::very_generic_py_unary_to_cpp_std_list_like` (PyObject *op, ListLike< T > &list_like)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_like_to_py_tuple` (const std::vector< T > &container)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_like_to_py_tuple` (const std::list< T > &container)
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_list_like` (PyObject *op, std::vector< T > &container)
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_list_like` (PyObject *op, std::list< T > &container)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_like_to_py_list` (const std::vector< T > &container)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_like_to_py_list` (const std::list< T > &container)
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_list_to_cpp_std_list_like` (PyObject *op, std::vector< T > &container)
- `template<typename T , int (*)(PyObject *) PyObject_Check, T (*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_list_to_cpp_std_list_like` (PyObject *op, std::list< T > &container)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy, PyObject (*)(PyObject *) PyContainer_New>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_unordered_set_to_py_set_or_frozenset` (const std::unordered_set< T > &set)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_unordered_set_to_py_set` (const std::unordered_set< T > &set)
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_unordered_set_to_py_frozenset` (const std::unordered_set< T > &set)

- `template<typename T , int(*)(PyObject *) PyObject_Check, int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_set_or_frozenset_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &set)`
- `template<typename T , int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_set_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &set)`
- `template<typename T , int(*)(PyObject *) PyObject_Check, T(*)(PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_frozenset_to_cpp_std_unordered_set (PyObject *op, std::unordered_set< T > &set)`
- `template<template< typename ... > class Map, typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V &) Convert_V>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_map_like_to_py_dict (const Map< K, V > &map)`
- `template<template< typename ... > class Map, typename K , typename V , int(*)(PyObject *) Check_K, int(*)(PyObject *) Check_V, K(*)(PyObject *) Convert_K, V(*)(PyObject *) Convert_V>`
`int Python_Cpp_Containers::generic_py_dict_to_cpp_std_map_like (PyObject *dict, Map< K, V > &map)`

9.12.1 Macro Definition Documentation

9.12.1.1 PYTHON_CPP_CONTAINERS_VERSION

```
#define PYTHON_CPP_CONTAINERS_VERSION "0.3.2"
```

Conversion functions for individual Python objects.

python_convert.hpp PythonC++

This contains hand crafted conversion of C++ <-> Python containers of homogeneous types. These are further instantiated by auto-generated code for a specific cartesian product of types and containers. That product is controlled by [code_gen.py](#) That auto-generated file is included by `#include "auto_py_convert_internal.h"` at the end of this file.

Created by Paul Ross on 22/11/2018. Copyright © 2018 Paul Ross. All rights reserved. Functions to handle Python containers.

9.13 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_convert_scrap.h File Reference

```
#include <Python.h>
#include <list>
#include <map>
#include <unordered_map>
#include <vector>
#include "python_object_convert.h"
#include "python_container_convert.h"
```

Namespaces

- [Python_Cpp_Containers](#)

Conversion functions for individual Python objects.

Functions

- `template<template< typename ... > class ListLike, typename T , PyObject (*)(const T &) ConvertCppToPy, PyObject (*)(size_t) PyUnaryContainer_New, int(*) (PyObject *, size_t, PyObject *) PyUnaryContainer_Set>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_like_to_py_list_like (const ListLike< T > &list_like)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_vector_to_py_tuple (const std::vector< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_to_py_tuple (const std::list< T > &container)`
- `template<typename T >`
`PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple (const std::vector< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (const std::vector< long > &container)`
- `template<typename T >`
`PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple (const std::list< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_tuple< long > (const std::list< long > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_vector_to_py_list (const std::vector< T > &container)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_list_to_py_list (const std::list< T > &container)`
- `template<typename T >`
`PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list (const std::vector< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (const std::vector< long > &container)`
- `template<typename T >`
`PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list (const std::list< T > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< long > (const std::list< long > &container)`
- `template<template< typename ... > class ListLike, typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert, int(*) (PyObject *) PyUnaryContainer_Check, Py_ssize_t(*) (PyObject *) PyUnaryContainer_Size, PyObject (*)(PyObject *, size_t) PyUnaryContainer_Get>`
`int Python_Cpp_Containers::generic_py_unary_to_cpp_std_list_like (PyObject *op, ListLike< T > &list_like)`
- `template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_vector (PyObject *op, std::vector< T > &vec)`
- `template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_list_to_cpp_std_vector (PyObject *op, std::vector< T > &vec)`
- `template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_tuple_to_cpp_std_list (PyObject *op, std::list< T > &vec)`
- `template<typename T , int(*) (PyObject *) PyObject_Check, T(*) (PyObject *) PyObject_Convert>`
`int Python_Cpp_Containers::generic_py_list_to_cpp_std_list (PyObject *op, std::list< T > &vec)`
- `template<typename T >`
`int Python_Cpp_Containers::py_list_to_cpp_std_list_like (PyObject *op, std::vector< T > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (PyObject *op, std::vector< long > &container)`
- `template<typename T >`
`int Python_Cpp_Containers::py_list_to_cpp_std_list_like (PyObject *op, std::list< T > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< long > (PyObject *op, std::list< long > &container)`

- `template<template< typename ... > class Map, typename K, typename V, PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>`
`PyObject * Python_Cpp_Containers::generic_cpp_std_map_like_to_py_dict (const Map< K, V > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict (const std::unordered_map< K, V > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::unordered_map, long, long >`
`(const std::unordered_map< long, long > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict (const std::map< K, V > &map)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, long >`
`(const std::map< long, long > &map)`
- `template<template< typename ... > class Map, typename K, typename V, int (*)(PyObject *) Check_K, int (*)(PyObject *) Check_V,`
`K (*)(PyObject *) Convert_K, V (*)(PyObject *) Convert_V>`
`int Python_Cpp_Containers::generic_py_dict_to_cpp_std_map_like (PyObject *dict, Map< K, V > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`int Python_Cpp_Containers::py_dict_to_cpp_std_map_like (PyObject *op, std::unordered_map< K, V > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::unordered_map, long, long >`
`(PyObject *op, std::unordered_map< long, long > &map)`
- `template<template< typename ... > class Map, typename K, typename V >`
`int Python_Cpp_Containers::py_dict_to_cpp_std_map_like (PyObject *op, std::map< K, V > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, long > (PyObject`
`*op, std::map< long, long > &map)`

9.14 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_convert.cpp File Reference

```
#include "python_object_convert.h"
```

Namespaces

- [Python_Cpp_Containers](#)
Conversion functions for individual Python objects.

Functions

- `PyObject * Python_Cpp_Containers::cpp_bool_to_py_bool (bool const &b)`
- `bool Python_Cpp_Containers::py_bool_to_cpp_bool (PyObject *op)`
- `int Python_Cpp_Containers::py_bool_check (PyObject *op)`
- `PyObject * Python_Cpp_Containers::cpp_long_to_py_long (const long &l)`
- `long Python_Cpp_Containers::py_long_to_cpp_long (PyObject *op)`
- `int Python_Cpp_Containers::py_long_check (PyObject *op)`
- `PyObject * Python_Cpp_Containers::cpp_double_to_py_float (const double &d)`
- `double Python_Cpp_Containers::py_float_to_cpp_double (PyObject *op)`
- `int Python_Cpp_Containers::py_float_check (PyObject *op)`
- `PyObject * Python_Cpp_Containers::cpp_complex_to_py_complex (const std::complex< double > &c)`
- `std::complex< double > Python_Cpp_Containers::py_complex_to_cpp_complex (PyObject *op)`

- int [Python_Cpp_Containers::py_complex_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_vector_char_to_py_bytes](#) (const std::vector< char > &s)
- std::vector< char > [Python_Cpp_Containers::py_bytes_to_cpp_vector_char](#) (PyObject *op)
- int [Python_Cpp_Containers::py_bytes_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_string_to_py_bytearray](#) (const std::vector< char > &s)
- std::vector< char > [Python_Cpp_Containers::py_bytearray_to_cpp_string](#) (PyObject *op)
- int [Python_Cpp_Containers::py_bytearray_check](#) (PyObject *op)
- int [Python_Cpp_Containers::py_unicode8_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_string_to_py_unicode8](#) (const std::string &s)
- std::string [Python_Cpp_Containers::py_unicode8_to_cpp_string](#) (PyObject *op)
- int [Python_Cpp_Containers::py_unicode16_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_u16string_to_py_unicode16](#) (const std::u16string &s)
- std::u16string [Python_Cpp_Containers::py_unicode16_to_cpp_u16string](#) (PyObject *op)
- int [Python_Cpp_Containers::py_unicode32_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_u32string_to_py_unicode32](#) (const std::u32string &s)
- std::u32string [Python_Cpp_Containers::py_unicode32_to_cpp_u32string](#) (PyObject *op)

9.15 /Users/paulross/CLionProjects/PythonCppHomogeneous↵ Containers/src/cpy/python_object_convert.h File Reference

```
#include <Python.h>
#include <complex>
#include <string>
#include <vector>
```

Namespaces

- [Python_Cpp_Containers](#)
Conversion functions for individual Python objects.

Functions

- PyObject * [Python_Cpp_Containers::cpp_bool_to_py_bool](#) (bool const &b)
- bool [Python_Cpp_Containers::py_bool_to_cpp_bool](#) (PyObject *op)
- int [Python_Cpp_Containers::py_bool_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_long_to_py_long](#) (const long &l)
- long [Python_Cpp_Containers::py_long_to_cpp_long](#) (PyObject *op)
- int [Python_Cpp_Containers::py_long_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_double_to_py_float](#) (const double &d)
- double [Python_Cpp_Containers::py_float_to_cpp_double](#) (PyObject *op)
- int [Python_Cpp_Containers::py_float_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_complex_to_py_complex](#) (const std::complex< double > &c)
- std::complex< double > [Python_Cpp_Containers::py_complex_to_cpp_complex](#) (PyObject *op)
- int [Python_Cpp_Containers::py_complex_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_vector_char_to_py_bytes](#) (const std::vector< char > &s)
- std::vector< char > [Python_Cpp_Containers::py_bytes_to_cpp_vector_char](#) (PyObject *op)
- int [Python_Cpp_Containers::py_bytes_check](#) (PyObject *op)
- PyObject * [Python_Cpp_Containers::cpp_vector_char_to_py_bytearray](#) (const std::vector< char > &s)

- `std::vector< char > Python_Cpp_Containers::py_bytearray_to_cpp_vector_char` (PyObject *op)
- `int Python_Cpp_Containers::py_bytearray_check` (PyObject *op)
- `int Python_Cpp_Containers::py_unicode8_check` (PyObject *op)
- `PyObject * Python_Cpp_Containers::cpp_string_to_py_unicode8` (const std::string &s)
- `std::string Python_Cpp_Containers::py_unicode8_to_cpp_string` (PyObject *op)
- `int Python_Cpp_Containers::py_unicode16_check` (PyObject *op)
- `PyObject * Python_Cpp_Containers::cpp_u16string_to_py_unicode16` (const std::u16string &s)
- `std::u16string Python_Cpp_Containers::py_unicode16_to_cpp_u16string` (PyObject *op)
- `int Python_Cpp_Containers::py_unicode32_check` (PyObject *op)
- `PyObject * Python_Cpp_Containers::cpp_u32string_to_py_unicode32` (const std::u32string &s)
- `std::u32string Python_Cpp_Containers::py_unicode32_to_cpp_u32string` (PyObject *op)

9.16 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.cpp File Reference

```
#include "test_common.h"
```

Functions

- `template<> int compare_tuple< bool >` (const std::vector< bool > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< long >` (const std::vector< long > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< double >` (const std::vector< double > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< std::complex< double > >` (const std::vector< std::complex< double > > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< std::vector< char > >` (const std::vector< std::vector< char > > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< std::string >` (const std::vector< std::string > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< std::u16string >` (const std::vector< std::u16string > &cpp_vector, PyObject *op)
- `template<> int compare_tuple< std::u32string >` (const std::vector< std::u32string > &cpp_vector, PyObject *op)
- `template<> int compare_list< bool >` (const std::vector< bool > &cpp_vector, PyObject *op)
- `template<> int compare_list< long >` (const std::vector< long > &cpp_vector, PyObject *op)
- `template<> int compare_list< double >` (const std::vector< double > &cpp_vector, PyObject *op)
- `template<> int compare_list< std::complex< double > >` (const std::vector< std::complex< double > > &cpp_vector, PyObject *op)
- `template<> int compare_list< std::vector< char > >` (const std::vector< std::vector< char > > &cpp_vector, PyObject *op)
- `template<> int compare_list< std::string >` (const std::vector< std::string > &cpp_vector, PyObject *op)
- `template<> int compare_list< std::u16string >` (const std::vector< std::u16string > &cpp_vector, PyObject *op)
- `template<> int compare_list< std::u32string >` (const std::vector< std::u32string > &cpp_vector, PyObject *op)
- `template<> int compare_set< std::vector< char > >` (const std::unordered_set< std::vector< char > > &cpp_set, PyObject *op)
- `template<> int compare_set< std::string >` (const std::unordered_set< std::string > &cpp_set, PyObject *op)
- `template<> int compare_set< std::u16string >` (const std::unordered_set< std::u16string > &cpp_set, PyObject *op)

- `template<> int compare_set< std::u32string > (const std::unordered_set< std::u32string > &cpp_set, PyObject *op)`
- `template<> int compare_dict< std::unordered_map, std::string, std::string > (const std::unordered_map< std::string, std::string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::unordered_map, std::u16string, std::u16string > (const std::unordered_map< std::u16string, std::u16string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::unordered_map, std::u32string, std::u32string > (const std::unordered_map< std::u32string, std::u32string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::map, std::string, std::string > (const std::map< std::string, std::string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::map, std::u16string, std::u16string > (const std::map< std::u16string, std::u16string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::map, std::u32string, std::u32string > (const std::map< std::u32string, std::u32string > &cpp_map, PyObject *op)`
- `int test_vector_vector_char_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_bytes_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_string_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_str_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u16string_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_str16_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u32string_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_str32_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_vector_char_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_list_bytes_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_string_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_list_str_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u16string_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_list_str16_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u32string_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_list_str32_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_bytes_to_py_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_set_bytes_to_unordered_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_string_to_py_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_set_string_to_unordered_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_u16string_to_py_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_set_string16_to_unordered_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_u32string_to_py_set (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_set_string32_to_unordered_set (TestResultS &test_results, size_t size, size_t str_len)`
- `PyObject * new_py_tuple_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_tuple_string (size_t size, size_t str_len)`
- `PyObject * new_py_tuple_string16 (size_t size, size_t str_len)`
- `PyObject * new_py_tuple_string32 (size_t size, size_t str_len)`
- `PyObject * new_py_list_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_list_string (size_t size, size_t str_len)`
- `PyObject * new_py_list_string16 (size_t size, size_t str_len)`
- `PyObject * new_py_list_string32 (size_t size, size_t str_len)`
- `PyObject * new_py_set_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_set_string (size_t size, size_t str_len)`
- `PyObject * new_py_set_u16string (size_t size, size_t str_len)`
- `PyObject * new_py_set_u32string (size_t size, size_t str_len)`
- `PyObject * new_py_dict_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_dict_string (size_t size, size_t str_len)`
- `PyObject * new_py_dict_string16 (size_t size, size_t str_len)`
- `PyObject * new_py_dict_string32 (size_t size, size_t str_len)`

- `template<template< typename ... > class MapLike>`
`int test_cpp_std_map_like_to_py_dict_bytes (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `int test_cpp_std_unordered_map_to_py_dict_bytes (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_bytes (TestResultS &test_results, size_t size, size_t str_len)`
- `template<template< typename ... > class MapLike>`
`int test_py_dict_to_cpp_std_map_like_bytes (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `int test_py_dict_to_cpp_std_unordered_map_bytes (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_bytes (TestResultS &test_results, size_t size, size_t str_len)`
- `template<template< typename ... > class MapLike>`
`int test_cpp_std_map_like_to_py_dict_string (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `template<template< typename ... > class MapLike>`
`int test_cpp_std_map_like_to_py_dict_string16 (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `template<template< typename ... > class MapLike>`
`int test_cpp_std_map_like_to_py_dict_string32 (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `int test_cpp_std_unordered_map_to_py_dict_string (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_unordered_map_to_py_dict_string16 (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_unordered_map_to_py_dict_string32 (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_string (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_string16 (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_string32 (TestResultS &test_results, size_t size, size_t str_len)`
- `template<template< typename ... > class MapLike>`
`int test_py_dict_to_cpp_std_map_like_string (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `template<template< typename ... > class MapLike>`
`int test_py_dict_to_cpp_std_map_like_string16 (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `template<template< typename ... > class MapLike>`
`int test_py_dict_to_cpp_std_map_like_string32 (TestResultS &test_results, size_t size, size_t str_len, const std::string &container_type)`
- `int test_py_dict_to_cpp_std_unordered_map_string (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_unordered_map_u16string (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_unordered_map_u32string (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_string (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_string16 (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_string32 (TestResultS &test_results, size_t size, size_t str_len)`

9.16.1 Function Documentation

9.16.1.1 `compare_dict< std::map, std::string, std::string >()`

```
template<>
int compare_dict< std::map, std::string, std::string > (
    const std::map< std::string, std::string > & cpy_map,
    PyObject * op )
```

9.16.1.2 `compare_dict< std::map, std::u16string, std::u16string >()`

```
template<>
int compare_dict< std::map, std::u16string, std::u16string > (
    const std::map< std::u16string, std::u16string > & cpp_map,
    PyObject * op )
```

9.16.1.3 `compare_dict< std::map, std::u32string, std::u32string >()`

```
template<>
int compare_dict< std::map, std::u32string, std::u32string > (
    const std::map< std::u32string, std::u32string > & cpp_map,
    PyObject * op )
```

9.16.1.4 `compare_dict< std::unordered_map, std::string, std::string >()`

```
template<>
int compare_dict< std::unordered_map, std::string, std::string > (
    const std::unordered_map< std::string, std::string > & cpp_map,
    PyObject * op )
```

9.16.1.5 `compare_dict< std::unordered_map, std::u16string, std::u16string >()`

```
template<>
int compare_dict< std::unordered_map, std::u16string, std::u16string > (
    const std::unordered_map< std::u16string, std::u16string > & cpp_map,
    PyObject * op )
```

9.16.1.6 `compare_dict< std::unordered_map, std::u32string, std::u32string >()`

```
template<>
int compare_dict< std::unordered_map, std::u32string, std::u32string > (
    const std::unordered_map< std::u32string, std::u32string > & cpp_map,
    PyObject * op )
```

9.16.1.7 `compare_list< bool >()`

```
template<>
int compare_list< bool > (
    const std::vector< bool > & cpp_vector,
    PyObject * op )
```


9.16.1.8 `compare_list< double >()`

```
template<>
int compare_list< double > (
    const std::vector< double > & cpp_vector,
    PyObject * op )
```

9.16.1.9 `compare_list< long >()`

```
template<>
int compare_list< long > (
    const std::vector< long > & cpp_vector,
    PyObject * op )
```

9.16.1.10 `compare_list< std::complex< double > >()`

```
template<>
int compare_list< std::complex< double > > (
    const std::vector< std::complex< double >> & cpp_vector,
    PyObject * op )
```

9.16.1.11 `compare_list< std::string >()`

```
template<>
int compare_list< std::string > (
    const std::vector< std::string > & cpp_vector,
    PyObject * op )
```

9.16.1.12 `compare_list< std::u16string >()`

```
template<>
int compare_list< std::u16string > (
    const std::vector< std::u16string > & cpp_vector,
    PyObject * op )
```

9.16.1.13 `compare_list< std::u32string >()`

```
template<>
int compare_list< std::u32string > (
    const std::vector< std::u32string > & cpp_vector,
    PyObject * op )
```

9.16.1.14 compare_list< std::vector< char > >()

```
template<>
int compare_list< std::vector< char > > (
    const std::vector< std::vector< char >> & cpp_vector,
    PyObject * op )
```

9.16.1.15 compare_set< std::string >()

```
template<>
int compare_set< std::string > (
    const std::unordered_set< std::string > & cpp_set,
    PyObject * op )
```

9.16.1.16 compare_set< std::u16string >()

```
template<>
int compare_set< std::u16string > (
    const std::unordered_set< std::u16string > & cpp_set,
    PyObject * op )
```

9.16.1.17 compare_set< std::u32string >()

```
template<>
int compare_set< std::u32string > (
    const std::unordered_set< std::u32string > & cpp_set,
    PyObject * op )
```

9.16.1.18 compare_set< std::vector< char > >()

```
template<>
int compare_set< std::vector< char > > (
    const std::unordered_set< std::vector< char >> & cpp_set,
    PyObject * op )
```

9.16.1.19 compare_tuple< bool >()

```
template<>
int compare_tuple< bool > (
    const std::vector< bool > & cpp_vector,
    PyObject * op )
```

9.16.1.20 `compare_tuple< double >()`

```
template<>
int compare_tuple< double > (
    const std::vector< double > & cpp_vector,
    PyObject * op )
```

9.16.1.21 `compare_tuple< long >()`

```
template<>
int compare_tuple< long > (
    const std::vector< long > & cpp_vector,
    PyObject * op )
```

9.16.1.22 `compare_tuple< std::complex< double > >()`

```
template<>
int compare_tuple< std::complex< double > > (
    const std::vector< std::complex< double >> & cpp_vector,
    PyObject * op )
```

9.16.1.23 `compare_tuple< std::string >()`

```
template<>
int compare_tuple< std::string > (
    const std::vector< std::string > & cpp_vector,
    PyObject * op )
```

9.16.1.24 `compare_tuple< std::u16string >()`

```
template<>
int compare_tuple< std::u16string > (
    const std::vector< std::u16string > & cpp_vector,
    PyObject * op )
```

9.16.1.25 `compare_tuple< std::u32string >()`

```
template<>
int compare_tuple< std::u32string > (
    const std::vector< std::u32string > & cpp_vector,
    PyObject * op )
```

9.16.1.26 `compare_tuple< std::vector< char > >()`

```
template<>
int compare_tuple< std::vector< char > > (
    const std::vector< std::vector< char >> & cpp_vector,
    PyObject * op )
```

9.16.1.27 `new_py_dict_bytes()`

```
PyObject* new_py_dict_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python dict of bytes for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each bytes object to be the key and value. Each key and value will be unique.

Returns

New reference to a dict or NULL on failure.

9.16.1.28 `new_py_dict_string()`

```
PyObject* new_py_dict_string (
    size_t size,
    size_t str_len )
```

Create a new Python dict of str for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each str object to be the key and value. Each key and value will be unique.

Returns

New reference to a dict or NULL on failure.

9.16.1.29 new_py_dict_string16()

```
PyObject* new_py_dict_string16 (  
    size_t size,  
    size_t str_len )
```

Create a new Python dict of `str` (16bit) for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each <code>str</code> object to be the key and value. Each key and value will be unique.

Returns

New reference to a dict or NULL on failure.

9.16.1.30 new_py_dict_string32()

```
PyObject* new_py_dict_string32 (  
    size_t size,  
    size_t str_len )
```

Create a new Python dict of `str` (32bit) for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each <code>str</code> object to be the key and value. Each key and value will be unique.

Returns

New reference to a dict or NULL on failure.

9.16.1.31 new_py_list_bytes()

```
PyObject* new_py_list_bytes (  
    size_t size,  
    size_t str_len )
```

Create a new Python list of bytes.

Parameters

<i>size</i>	Length of the list.
<i>str_len</i>	Length of each <code>bytes</code> object. Each byte is just <code>''</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.16.1.32 new_py_list_string()

```
PyObject* new_py_list_string (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `str`.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each byte is just <code>'</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.16.1.33 new_py_list_string16()

```
PyObject* new_py_list_string16 (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `str` with 16 bit characters.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just <code>u'</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.16.1.34 new_py_list_string32()

```
PyObject* new_py_list_string32 (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `str` with 32 bit characters.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just U' '.

Returns

New reference to a `list` or `NULL` on failure.

9.16.1.35 new_py_set_bytes()

```
PyObject* new_py_set_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python `set` of `bytes`.

Parameters

<i>size</i>	Length of the <code>set</code> .
<i>str_len</i>	Length of each <code>bytes</code> object. Each object will be unique.

Returns

New reference to a `set` or `NULL` on failure.

9.16.1.36 new_py_set_string()

```
PyObject* new_py_set_string (
    size_t size,
    size_t str_len )
```

Create a new Python `set` of `str`.

Parameters

<i>size</i>	Length of the <code>set</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each object will be unique.

Returns

New reference to a `set` or `NULL` on failure.

9.16.1.37 new_py_set_u16string()

```
PyObject* new_py_set_u16string (
    size_t size,
    size_t str_len )
```

Create a new Python set of `str` 16 bit.

Parameters

<i>size</i>	Length of the set.
<i>str_len</i>	Length of each <code>str</code> object. Each object will be unique.

Returns

New reference to a `set` or `NULL` on failure.

9.16.1.38 new_py_set_u32string()

```
PyObject* new_py_set_u32string (
    size_t size,
    size_t str_len )
```

Create a new Python set of `str` 32 bit.

Parameters

<i>size</i>	Length of the set.
<i>str_len</i>	Length of each <code>str</code> object. Each object will be unique.

Returns

New reference to a `set` or `NULL` on failure.

9.16.1.39 new_py_tuple_bytes()

```
PyObject* new_py_tuple_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python tuple of bytes.

Parameters

<i>size</i>	Length of the tuple.
<i>str_len</i>	Length of each <code>bytes</code> object. Each byte is just <code>' '</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.16.1.40 new_py_tuple_string()

```
PyObject* new_py_tuple_string (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of `str`.

Parameters

<i>size</i>	Length of the <code>tuple</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each byte is just <code>'</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.16.1.41 new_py_tuple_string16()

```
PyObject* new_py_tuple_string16 (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of 16 bit unicode `str`.

Parameters

<i>size</i>	Length of the <code>tuple</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just <code>u'</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.16.1.42 new_py_tuple_string32()

```
PyObject* new_py_tuple_string32 (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of 32 bit unicode `str`.

Parameters

<i>size</i>	Length of the tuple.
<i>str_len</i>	Length of each <code>str</code> object. Each character is just U' '.

Returns

New reference to a tuple or NULL on failure.

9.16.1.43 test_cpp_std_map_like_to_py_dict_bytes()

```
template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.44 test_cpp_std_map_like_to_py_dict_string()

```
template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_string (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.45 test_cpp_std_map_like_to_py_dict_string16()

```
template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.46 test_cpp_std_map_like_to_py_dict_string32()

```
template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.47 test_cpp_std_map_to_py_dict_bytes()

```
int test_cpp_std_map_to_py_dict_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.48 test_cpp_std_map_to_py_dict_string()

```
int test_cpp_std_map_to_py_dict_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.49 test_cpp_std_map_to_py_dict_string16()

```
int test_cpp_std_map_to_py_dict_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.50 test_cpp_std_map_to_py_dict_string32()

```
int test_cpp_std_map_to_py_dict_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.51 test_cpp_std_unordered_map_to_py_dict_bytes()

```
int test_cpp_std_unordered_map_to_py_dict_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.52 test_cpp_std_unordered_map_to_py_dict_string()

```
int test_cpp_std_unordered_map_to_py_dict_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.53 test_cpp_std_unordered_map_to_py_dict_string16()

```
int test_cpp_std_unordered_map_to_py_dict_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.54 test_cpp_std_unordered_map_to_py_dict_string32()

```
int test_cpp_std_unordered_map_to_py_dict_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.55 test_py_dict_to_cpp_std_map_bytes()

```
int test_py_dict_to_cpp_std_map_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.56 test_py_dict_to_cpp_std_map_like_bytes()

```
template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.57 test_py_dict_to_cpp_std_map_like_string()

```
template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_string (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.58 test_py_dict_to_cpp_std_map_like_string16()

```
template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.59 test_py_dict_to_cpp_std_map_like_string32()

```
template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    const std::string & container_type )
```

9.16.1.60 test_py_dict_to_cpp_std_map_string()

```
int test_py_dict_to_cpp_std_map_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.61 test_py_dict_to_cpp_std_map_string16()

```
int test_py_dict_to_cpp_std_map_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.62 test_py_dict_to_cpp_std_map_string32()

```
int test_py_dict_to_cpp_std_map_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.63 test_py_dict_to_cpp_std_unordered_map_bytes()

```
int test_py_dict_to_cpp_std_unordered_map_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.64 test_py_dict_to_cpp_std_unordered_map_string()

```
int test_py_dict_to_cpp_std_unordered_map_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.65 test_py_dict_to_cpp_std_unordered_map_u16string()

```
int test_py_dict_to_cpp_std_unordered_map_u16string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.66 test_py_dict_to_cpp_std_unordered_map_u32string()

```
int test_py_dict_to_cpp_std_unordered_map_u32string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.67 test_py_list_bytes_to_vector()

```
int test_py_list_bytes_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python list to a C++ `std::vector<std::vector<char>>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.68 test_py_list_str16_to_vector()

```
int test_py_list_str16_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python list to a C++ `std::vector<std::u16string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.69 test_py_list_str32_to_vector()

```
int test_py_list_str32_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python list to a C++ `std::vector<std::u32string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.70 test_py_list_str_to_vector()

```
int test_py_list_str_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python list to a C++ `std::vector<std::string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.71 test_py_set_bytes_to_unordered_set()

```
int test_py_set_bytes_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.72 test_py_set_string16_to_unordered_set()

```
int test_py_set_string16_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.73 test_py_set_string32_to_unordered_set()

```
int test_py_set_string32_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```


9.16.1.74 test_py_set_string_to_unordered_set()

```
int test_py_set_string_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.75 test_py_tuple_bytes_to_vector()

```
int test_py_tuple_bytes_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of bytes to a C++ `std::vector<std::vector<char>>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.76 test_py_tuple_str16_to_vector()

```
int test_py_tuple_str16_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of str to a C++ `std::vector<std::string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.77 test_py_tuple_str32_to_vector()

```
int test_py_tuple_str32_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of str to a C++ `std::vector<std::u32string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.78 test_py_tuple_str_to_vector()

```
int test_py_tuple_str_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of str to a C++ `std::vector<std::string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.79 test_unordered_set_bytes_to_py_set()

```
int test_unordered_set_bytes_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.80 test_unordered_set_string_to_py_set()

```
int test_unordered_set_string_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.81 test_unordered_set_u16string_to_py_set()

```
int test_unordered_set_u16string_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.82 test_unordered_set_u32string_to_py_set()

```
int test_unordered_set_u32string_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.16.1.83 test_vector_string_to_py_list()

```
int test_vector_string_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::string>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.84 test_vector_string_to_py_tuple()

```
int test_vector_string_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::string>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.85 test_vector_u16string_to_py_list()

```
int test_vector_u16string_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u16string>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.86 test_vector_u16string_to_py_tuple()

```
int test_vector_u16string_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u16string>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.87 test_vector_u32string_to_py_list()

```
int test_vector_u32string_to_py_list (  
    TestResults & test_results,  
    size_t size,  
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u32string>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.88 test_vector_u32string_to_py_tuple()

```
int test_vector_u32string_to_py_tuple (  
    TestResults & test_results,  
    size_t size,  
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u32string>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.89 test_vector_vector_char_to_py_list()

```
int test_vector_vector_char_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::vector<char>>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.16.1.90 test_vector_vector_char_to_py_tuple()

```
int test_vector_vector_char_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::vector<char>>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.h File Reference

```
#include <Python.h>
#include "cpy/python_convert.h"
#include "cpy/get_rss.h"
#include "cpy/TestFramework.h"
```

Macros

- `#define REPORT_OK_OR_FAIL 0`
If non-zero then report the result of every test. This is very verbose.
- `#define REPORT_TEST_OUTPUT`
- `#define REPORT_TEST_OUTPUT_WITH_TYPE`
- `#define REPORT_TEST_OUTPUT_WITH_STRING_LENGTH`
- `#define REPORT_TEST_OUTPUT_WITH_CONTAINER_TYPE_STRING_LENGTH`
- `#define RSS_SNAPSHOT`
Enable the macros that do RSS snapshots.
- `#define RSS_SNAPSHOT_WITHOUT_TYPE RSSSnapshot rss(__FUNCTION__);`
- `#define RSS_SNAPSHOT_WITH_TYPE(type)`
- `#define RSS_SNAPSHOT_WITH_CONTAINER_TYPE_AND_TYPE(container_type, type)`
- `#define RSS_SNAPSHOT_REPORT std::cout << rss << std::endl;`
- `#define TEST_FOR_PY_ERR_ON_ENTRY`
- `#define TEST_FOR_PY_ERR_ON_EXIT`
- `#define SET_RESULT_IF_PY_ERR_OCCURRED`

Functions

- `template<typename T, PyObject (*)(const T &) Convert_T_To_Py, T (*)(PyObject *) Convert_Py_To_T, int (*)(PyObject *) PyUnaryContainer_Check, Py_ssize_t (*)(PyObject *) PyUnaryContainer_Size, PyObject (*)(PyObject *, size_t) PyUnaryContainer_Get> int compare_tuple_or_list (std::vector< T > const &cpp_vector, PyObject *op)`
- `template<typename T, PyObject (*)(const T &) Convert_T_To_Py, T (*)(PyObject *) Convert_Py_To_T> int compare_tuple (std::vector< T > const &cpp_vector, PyObject *op)`
- `template<typename T > int compare_tuple (const std::vector< T > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< bool > (const std::vector< bool > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< long > (const std::vector< long > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< double > (const std::vector< double > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< std::complex< double > > (const std::vector< std::complex< double > > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< std::vector< char > > (const std::vector< std::vector< char > > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< std::string > (const std::vector< std::string > &cpp_vector, PyObject *op)`
- `template<> int compare_tuple< std::u16string > (const std::vector< std::u16string > &cpp_vector, PyObject *op)`

- `template<> int compare_tuple< std::u32string > (const std::vector< std::u32string > &cpp_vector, PyObject *op)`
- `template<typename T , PyObject (*)(const T &) Convert_T_To_Py, T (*)(PyObject *) Convert_Py_To_T> int compare_list (std::vector< T > const &cpp_vector, PyObject *op)`
- `template<typename T > int compare_list (const std::vector< T > &cpp_vector, PyObject *op)`
- `template<> int compare_list< bool > (const std::vector< bool > &cpp_vector, PyObject *op)`
- `template<> int compare_list< long > (const std::vector< long > &cpp_vector, PyObject *op)`
- `template<> int compare_list< double > (const std::vector< double > &cpp_vector, PyObject *op)`
- `template<> int compare_list< std::complex< double > > (const std::vector< std::complex< double > > &cpp_vector, PyObject *op)`
- `template<> int compare_list< std::vector< char > > (const std::vector< std::vector< char > > &cpp_vector, PyObject *op)`
- `template<> int compare_list< std::string > (const std::vector< std::string > &cpp_vector, PyObject *op)`
- `template<typename T , PyObject (*)(const T &) Convert_T_To_Py, T (*)(PyObject *) Convert_Py_To_T> int compare_set (const std::unordered_set< T > &cpp_set, PyObject *op)`
- `template<typename T > int compare_set (const std::unordered_set< T > &cpp_set, PyObject *op)`
- `template<> int compare_set< std::vector< char > > (const std::unordered_set< std::vector< char > > &cpp_set, PyObject *op)`
- `template<> int compare_set< std::string > (const std::unordered_set< std::string > &cpp_set, PyObject *op)`
- `template<> int compare_set< std::u16string > (const std::unordered_set< std::u16string > &cpp_set, PyObject *op)`
- `template<> int compare_set< std::u32string > (const std::unordered_set< std::u32string > &cpp_set, PyObject *op)`
- `template<template< typename ... > class MapLike, typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V, K (*)(PyObject *) Convert_Py_Key, V (*)(PyObject *) Convert_Py_Val> int compare_dict (MapLike< K, V > const &cpp_map, PyObject *op)`
- `template<template< typename ... > class MapLike, typename K , typename V > int compare_dict (const MapLike< K, V > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::unordered_map, std::string, std::string > (const std::unordered_map< std::string, std::string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::unordered_map, std::u16string, std::u16string > (const std::unordered_map< std::u16string, std::u16string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::unordered_map, std::u32string, std::u32string > (const std::unordered_map< std::u32string, std::u32string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::map, std::string, std::string > (const std::map< std::string, std::string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::map, std::u16string, std::u16string > (const std::map< std::u16string, std::u16string > &cpp_map, PyObject *op)`
- `template<> int compare_dict< std::map, std::u32string, std::u32string > (const std::map< std::u32string, std::u32string > &cpp_map, PyObject *op)`
- `int test_vector_vector_char_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_bytes_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_string_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_str_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u16string_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_str16_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u32string_to_py_tuple (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_tuple_str32_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_vector_char_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_list_bytes_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_string_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_py_list_str_to_vector (TestResultS &test_results, size_t size, size_t str_len)`
- `int test_vector_u16string_to_py_list (TestResultS &test_results, size_t size, size_t str_len)`

- `int test_py_list_str16_to_vector (TestResults &test_results, size_t size, size_t str_len)`
- `int test_vector_u32string_to_py_list (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_list_str32_to_vector (TestResults &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_bytes_to_py_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_set_bytes_to_unordered_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_string_to_py_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_set_string_to_unordered_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_u16string_to_py_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_set_string16_to_unordered_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_unordered_set_u32string_to_py_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_set_string32_to_unordered_set (TestResults &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_unordered_map_to_py_dict_bytes (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_unordered_map_bytes (TestResults &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_unordered_map_to_py_dict_string (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_unordered_map_string (TestResults &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_unordered_map_to_py_dict_string16 (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_unordered_map_u16string (TestResults &test_results, size_t size, size_t str_↵ len)`
- `int test_cpp_std_unordered_map_to_py_dict_string32 (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_unordered_map_u32string (TestResults &test_results, size_t size, size_t str_↵ len)`
- `int test_cpp_std_map_to_py_dict_bytes (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_bytes (TestResults &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_string (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_string (TestResults &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_string16 (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_string16 (TestResults &test_results, size_t size, size_t str_len)`
- `int test_cpp_std_map_to_py_dict_string32 (TestResults &test_results, size_t size, size_t str_len)`
- `int test_py_dict_to_cpp_std_map_string32 (TestResults &test_results, size_t size, size_t str_len)`
- `template<typename T , T(*) (PyObject *) ConvertPyToCpp>`
`int test_vector_to_py_tuple (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T , PyObject *(*) (const T &) ConvertCppToPy>`
`int test_py_tuple_to_vector (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T >`
`int test_vector_to_py_tuple_round_trip (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T , PyObject *(*) (const T &) ConvertCppToPy>`
`int test_py_tuple_to_vector_round_trip (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T >`
`int test_vector_to_py_list (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T , PyObject *(*) (const T &) ConvertCppToPy, T(*) (PyObject *) ConvertPyToCpp>`
`int test_py_list_to_vector (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T >`
`int test_vector_to_py_list_round_trip (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T , PyObject *(*) (const T &) ConvertCppToPy>`
`int test_py_list_to_vector_round_trip (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T , T(*) (PyObject *) ConvertPyToCpp, PyObject *(*) (const T &) ConvertCppToPy>`
`int test_unordered_set_to_py_set (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename T , T(*) (PyObject *) ConvertPyToCpp, PyObject *(*) (const T &) ConvertCppToPy>`
`int test_py_set_to_unordered_set (TestResults &test_results, const std::string &type, size_t size)`
- `template<template< typename ... > class MapLike, typename K , typename V , PyObject *(*) (const K &) Convert_K, PyObject *(*) (const V &) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObject *) Convert_Py_Val>`
`int test_cpp_std_map_like_to_py_dict (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename K , typename V , PyObject *(*) (const K &) Convert_K, PyObject *(*) (const V &) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObject *) Convert_Py_Val>`
`int test_cpp_std_unordered_map_to_py_dict (TestResults &test_results, const std::string &type, size_t size)`

- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V, K (*)(PyObject *) Convert_Py_Key, V (*)(PyObject *) Convert_Py_Val>`
`int test_cpp_std_map_to_py_dict (TestResults &test_results, const std::string &type, size_t size)`
- `template<template< typename ... > class MapLike, typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V, K (*)(PyObject *) Convert_Py_Key, V (*)(PyObject *) Convert_Py_Val>`
`int test_py_dict_to_cpp_std_map_like (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V, K (*)(PyObject *) Convert_Py_Key, V (*)(PyObject *) Convert_Py_Val>`
`int test_py_dict_to_cpp_std_unordered_map (TestResults &test_results, const std::string &type, size_t size)`
- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V, K (*)(PyObject *) Convert_Py_Key, V (*)(PyObject *) Convert_Py_Val>`
`int test_py_dict_to_cpp_std_map (TestResults &test_results, const std::string &type, size_t size)`
- `PyObject * new_py_tuple_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_tuple_string (size_t size, size_t str_len)`
- `PyObject * new_py_tuple_string16 (size_t size, size_t str_len)`
- `PyObject * new_py_tuple_string32 (size_t size, size_t str_len)`
- `PyObject * new_py_list_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_list_string (size_t size, size_t str_len)`
- `PyObject * new_py_list_string16 (size_t size, size_t str_len)`
- `PyObject * new_py_list_string32 (size_t size, size_t str_len)`
- `PyObject * new_py_set_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_set_string (size_t size, size_t str_len)`
- `PyObject * new_py_set_u16string (size_t size, size_t str_len)`
- `PyObject * new_py_set_u32string (size_t size, size_t str_len)`
- `PyObject * new_py_dict_bytes (size_t size, size_t str_len)`
- `PyObject * new_py_dict_string (size_t size, size_t str_len)`
- `PyObject * new_py_dict_string16 (size_t size, size_t str_len)`
- `PyObject * new_py_dict_string32 (size_t size, size_t str_len)`

Variables

- `const int PY_ERR_ON_ENTRY_RETURN_CODE = -1`
- `const int PY_ERR_ON_EXIT_RETURN_CODE = -2`

9.17.1 Macro Definition Documentation

9.17.1.1 REPORT_OK_OR_FAIL

```
#define REPORT_OK_OR_FAIL 0
```

If non-zero then report the result of every test. This is very verbose.

9.17.1.2 REPORT_TEST_OUTPUT

```
#define REPORT_TEST_OUTPUT
```

Value:

```
do {
    std::ostringstream title; \
    title << __FUNCTION__ << "():" << "[" << size << ";" \
    test_results.push_back(测试结果(title.str(), result, exec_time, 1, size)); \
} while (0)
```

9.17.1.3 REPORT_TEST_OUTPUT_WITH_CONTAINER_TYPE_STRING_LENGTH

```
#define REPORT_TEST_OUTPUT_WITH_CONTAINER_TYPE_STRING_LENGTH
```

Value:

```
do {
    std::ostringstream title; \
    title << __FUNCTION__ << container_type << "<" << "std::string[" << str_len << ">" << ">" << "():" << "[" << size << ";" \
    test_results.push_back(测试结果(title.str(), result, exec_time, 1, size)); \
} while (0)
```

9.17.1.4 REPORT_TEST_OUTPUT_WITH_STRING_LENGTH

```
#define REPORT_TEST_OUTPUT_WITH_STRING_LENGTH
```

Value:

```
do {
    std::ostringstream title; \
    title << __FUNCTION__ << " std::string[" << str_len << ">" << "():" << "[" << size << ";" \
    test_results.push_back(测试结果(title.str(), result, exec_time, 1, size)); \
} while (0)
```

9.17.1.5 REPORT_TEST_OUTPUT_WITH_TYPE

```
#define REPORT_TEST_OUTPUT_WITH_TYPE
```

Value:

```
do {
    std::ostringstream title; \
    title << __FUNCTION__ << "<'<' << type << ">():" << "[" << size << ";" \
    test_results.push_back(测试结果(title.str(), result, exec_time, 1, size)); \
} while (0)
```

9.17.1.6 RSS_SNAPSHOT

```
#define RSS_SNAPSHOT
```

Enable the macros that do RSS snapshots.

9.17.1.7 RSS_SNAPSHOT_REPORT

```
#define RSS_SNAPSHOT_REPORT std::cout << rss << std::endl;
```

Report the RSS usage to `stdout`.

9.17.1.8 RSS_SNAPSHOT_WITH_CONTAINER_TYPE_AND_TYPE

```
#define RSS_SNAPSHOT_WITH_CONTAINER_TYPE_AND_TYPE(  
    container_type,  
    type )
```

Value:

```
std::ostringstream rss_title; \  
rss_title << __FUNCTION__ << " " << container_type << "<" << type << ">"; \  
RSSSnapshot rss(rss_title.str());
```

Take a snapshot of the current RSS value. The snapshot title is the function name followed by the container and type. For example if the function is "foo", the container is "vector" and the type is "long" the title will be "foo vector<long>".

9.17.1.9 RSS_SNAPSHOT_WITH_TYPE

```
#define RSS_SNAPSHOT_WITH_TYPE(  
    type )
```

Value:

```
std::ostringstream rss_title; \  
rss_title << __FUNCTION__ << type; \  
RSSSnapshot rss(rss_title.str());
```

Take a snapshot of the current RSS value. The snapshot title is the function name followed immediately by the type. For example if the function is "foo" and the type is "<long>" the title will be "foo<long>".

9.17.1.10 RSS_SNAPSHOT_WITHOUT_TYPE

```
#define RSS_SNAPSHOT_WITHOUT_TYPE RSSSnapshot rss(__FUNCTION__);
```

Take a snapshot of the current RSS value. The snapshot title is the function name.

9.17.1.11 SET_RESULT_IF_PY_ERR_OCCURRED

```
#define SET_RESULT_IF_PY_ERR_OCCURRED
```

Value:

```
do {  
    if (PyErr_Occurred()) {  
        PyErr_Print();  
        result = PY_ERR_ON_ENTRY_RETURN_CODE;  
    }  
} while (0)
```

9.17.1.12 TEST_FOR_PY_ERR_ON_ENTRY

```
#define TEST_FOR_PY_ERR_ON_ENTRY
```

Value:

```
do {
    if (PyErr_Occurred()) {
        PyErr_Print();
        return PY_ERR_ON_ENTRY_RETURN_CODE;
    }
} while(0)
```

9.17.1.13 TEST_FOR_PY_ERR_ON_EXIT

```
#define TEST_FOR_PY_ERR_ON_EXIT
```

Value:

```
do {
    if (PyErr_Occurred()) {
        PyErr_Print();
        return PY_ERR_ON_EXIT_RETURN_CODE;
    }
} while(0)
```

9.17.2 Function Documentation

9.17.2.1 compare_dict() [1/2]

```
template<template< typename ... > class MapLike, typename K , typename V >
int compare_dict (
    MapLike< K, V > const & cpp_map,
    PyObject * op )
```

Compare a Python dict with a C++ `std::unordered_map` or `std::map`.

Template Parameters

<i>MapLike</i>	The C++ type of the container.
<i>K</i>	The C++ type of the keys.
<i>V</i>	The C++ type of the values.
<i>Convert_K</i>	Pointer to function to convert a C++ type <i>K</i> to a <code>PyObject*</code> .
<i>Convert_V</i>	Pointer to function to convert a C++ type <i>V</i> to a <code>PyObject*</code> .
<i>Convert_Py_Key</i>	Pointer to function to convert a <code>PyObject*</code> key to a C++ type <i>K</i> .
<i>Convert_Py_Val</i>	Pointer to function to convert a <code>PyObject*</code> value to a C++ type <i>V</i> .

Parameters

<i>cpp_map</i>	The C++ <code>std::unordered_map</code> or <code>std::map</code> .
<i>op</i>	The Python dict.

Returns

0 if the same, non-zero if different.

9.17.2.2 compare_dict() [2/2]

```
template<template< typename ... > class MapLike, typename K , typename V , PyObject *(*)(const
K &) Convert_K, PyObject *(*)(const V &) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyOb↵
Object *) Convert_Py_Val>
int compare_dict (
    MapLike< K, V > const & cpp_map,
    PyObject * op )
```

Compare a Python dict with a C++ `std::unordered_map` or `std::map`.

Template Parameters

<i>MapLike</i>	The C++ type of the container.
<i>K</i>	The C++ type of the keys.
<i>V</i>	The C++ type of the values.
<i>Convert_K</i>	Pointer to function to convert a C++ type <i>K</i> to a <code>PyObject*</code> .
<i>Convert_V</i>	Pointer to function to convert a C++ type <i>V</i> to a <code>PyObject*</code> .
<i>Convert_Py_Key</i>	Pointer to function to convert a <code>PyObject*</code> key to a C++ type <i>K</i> .
<i>Convert_Py_Val</i>	Pointer to function to convert a <code>PyObject*</code> value to a C++ type <i>V</i> .

Parameters

<i>cpp_map</i>	The C++ <code>std::unordered_map</code> or <code>std::map</code> .
<i>op</i>	The Python dict.

Returns

0 if the same, non-zero if different.

9.17.2.3 compare_dict< std::map, std::string, std::string >()

```
template<>
int compare_dict< std::map, std::string, std::string > (
    const std::map< std::string, std::string > & cpp_map,
    PyObject * op )
```

9.17.2.4 `compare_dict< std::map, std::u16string, std::u16string >()`

```
template<>
int compare_dict< std::map, std::u16string, std::u16string > (
    const std::map< std::u16string, std::u16string > & cpp_map,
    PyObject * op )
```

9.17.2.5 `compare_dict< std::map, std::u32string, std::u32string >()`

```
template<>
int compare_dict< std::map, std::u32string, std::u32string > (
    const std::map< std::u32string, std::u32string > & cpp_map,
    PyObject * op )
```

9.17.2.6 `compare_dict< std::unordered_map, std::string, std::string >()`

```
template<>
int compare_dict< std::unordered_map, std::string, std::string > (
    const std::unordered_map< std::string, std::string > & cpp_map,
    PyObject * op )
```

9.17.2.7 `compare_dict< std::unordered_map, std::u16string, std::u16string >()`

```
template<>
int compare_dict< std::unordered_map, std::u16string, std::u16string > (
    const std::unordered_map< std::u16string, std::u16string > & cpp_map,
    PyObject * op )
```

9.17.2.8 `compare_dict< std::unordered_map, std::u32string, std::u32string >()`

```
template<>
int compare_dict< std::unordered_map, std::u32string, std::u32string > (
    const std::unordered_map< std::u32string, std::u32string > & cpp_map,
    PyObject * op )
```

9.17.2.9 `compare_list()` [1/2]

```
template<typename T >
int compare_list (
    std::vector< T > const & cpp_vector,
    PyObject * op )
```

Specialisation of `compare_tuple_or_list` that compares a Python list with a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type of objects in the vector.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type <i>T</i> to a <code>PyObject*</code> .
<i>Convert_Py_To_↔ _T</i>	Pointer to function to convert a <code>PyObject*</code> to a C++ type <i>T</i> .

Parameters

<i>cpp_vector</i>	The C++ <code>std::vector</code> .
<i>op</i>	The Python <code>list</code> .

Returns

0 if identical, non-zero if not.

9.17.2.10 compare_list() [2/2]

```
template<typename T , PyObject *(*)(const T &) Convert_T_To_Py, T(*) (PyObject *) Convert_Py_↔
To_T>
int compare_list (
    std::vector< T > const & cpp_vector,
    PyObject * op )
```

Specialisation of `compare_tuple_or_list` that compares a Python `list` with a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type of objects in the vector.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type <i>T</i> to a <code>PyObject*</code> .
<i>Convert_Py_To_↔ _T</i>	Pointer to function to convert a <code>PyObject*</code> to a C++ type <i>T</i> .

Parameters

<i>cpp_vector</i>	The C++ <code>std::vector</code> .
<i>op</i>	The Python <code>list</code> .

Returns

0 if identical, non-zero if not.

9.17.2.11 `compare_list< bool >()`

```
template<>
int compare_list< bool > (
    const std::vector< bool > & cpp_vector,
    PyObject * op )
```

9.17.2.12 `compare_list< double >()`

```
template<>
int compare_list< double > (
    const std::vector< double > & cpp_vector,
    PyObject * op )
```

9.17.2.13 `compare_list< long >()`

```
template<>
int compare_list< long > (
    const std::vector< long > & cpp_vector,
    PyObject * op )
```

9.17.2.14 `compare_list< std::complex< double > >()`

```
template<>
int compare_list< std::complex< double > > (
    const std::vector< std::complex< double >> & cpp_vector,
    PyObject * op )
```

9.17.2.15 `compare_list< std::string >()`

```
template<>
int compare_list< std::string > (
    const std::vector< std::string > & cpp_vector,
    PyObject * op )
```

9.17.2.16 `compare_list< std::vector< char > >()`

```
template<>
int compare_list< std::vector< char > > (
    const std::vector< std::vector< char >> & cpp_vector,
    PyObject * op )
```

9.17.2.17 compare_set() [1/2]

```
template<typename T , PyObject *(*)(const T &) Convert_T_To_Py, T(*) (PyObject *) Convert_Py_To_T>
int compare_set (
    const std::unordered_set< T > & cpp_set,
    PyObject * op )
```

Compares a Python set or frozenset with a C++ `std::unordered_set`.

Template Parameters

<i>T</i>	C++ type of objects in the set.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type T to a PyObject*.
<i>Convert_Py_To_T</i>	Pointer to function to convert a PyObject* to a C++ type T.

Parameters

<i>cpp_set</i>	The C++ <code>std::unordered_set</code> .
<i>op</i>	The Python set.

Returns

0 if identical, non-zero if not.

9.17.2.18 compare_set() [2/2]

```
template<typename T >
int compare_set (
    const std::unordered_set< T > & cpp_set,
    PyObject * op )
```

Compares a Python set or frozenset with a C++ `std::unordered_set`.

Template Parameters

<i>T</i>	C++ type of objects in the set.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type T to a PyObject*.
<i>Convert_Py_To_T</i>	Pointer to function to convert a PyObject* to a C++ type T.

Parameters

<i>cpp_set</i>	The C++ <code>std::unordered_set</code> .
<i>op</i>	The Python set.

Returns

0 if identical, non-zero if not.

9.17.2.19 compare_set< std::string >()

```
template<>
int compare_set< std::string > (
    const std::unordered_set< std::string > & cpp_set,
    PyObject * op )
```

9.17.2.20 compare_set< std::u16string >()

```
template<>
int compare_set< std::u16string > (
    const std::unordered_set< std::u16string > & cpp_set,
    PyObject * op )
```

9.17.2.21 compare_set< std::u32string >()

```
template<>
int compare_set< std::u32string > (
    const std::unordered_set< std::u32string > & cpp_set,
    PyObject * op )
```

9.17.2.22 compare_set< std::vector< char > >()

```
template<>
int compare_set< std::vector< char > > (
    const std::unordered_set< std::vector< char >> & cpp_set,
    PyObject * op )
```

9.17.2.23 compare_tuple() [1/2]

```
template<typename T >
int compare_tuple (
    std::vector< T > const & cpp_vector,
    PyObject * op )
```

Specialisation of `compare_tuple_or_list` that compares a Python tuple with a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type of objects in the vector.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type <i>T</i> to a <code>PyObject*</code> .
<i>Convert_Py_To_↔_T</i>	Pointer to function to convert a <code>PyObject*</code> to a C++ type <i>T</i> .

Parameters

<i>cpp_vector</i>	The C++ vector.
<i>op</i>	The Python tuple.

Returns

0 if identical, non-zero if not.

9.17.2.24 compare_tuple() [2/2]

```
template<typename T , PyObject *(*)(const T &) Convert_T_To_Py, T(*) (PyObject *) Convert_Py_↔_To_T>
int compare_tuple (
    std::vector< T > const & cpp_vector,
    PyObject * op )
```

Specialisation of `compare_tuple_or_list` that compares a Python tuple with a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type of objects in the vector.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type <i>T</i> to a <code>PyObject*</code> .
<i>Convert_Py_To_↔_T</i>	Pointer to function to convert a <code>PyObject*</code> to a C++ type <i>T</i> .

Parameters

<i>cpp_vector</i>	The C++ vector.
<i>op</i>	The Python tuple.

Returns

0 if identical, non-zero if not.

9.17.2.25 `compare_tuple< bool >()`

```
template<>
int compare_tuple< bool > (
    const std::vector< bool > & cpp_vector,
    PyObject * op )
```

9.17.2.26 `compare_tuple< double >()`

```
template<>
int compare_tuple< double > (
    const std::vector< double > & cpp_vector,
    PyObject * op )
```

9.17.2.27 `compare_tuple< long >()`

```
template<>
int compare_tuple< long > (
    const std::vector< long > & cpp_vector,
    PyObject * op )
```

9.17.2.28 `compare_tuple< std::complex< double > >()`

```
template<>
int compare_tuple< std::complex< double > > (
    const std::vector< std::complex< double >> & cpp_vector,
    PyObject * op )
```

9.17.2.29 `compare_tuple< std::string >()`

```
template<>
int compare_tuple< std::string > (
    const std::vector< std::string > & cpp_vector,
    PyObject * op )
```

9.17.2.30 `compare_tuple< std::u16string >()`

```
template<>
int compare_tuple< std::u16string > (
    const std::vector< std::u16string > & cpp_vector,
    PyObject * op )
```

9.17.2.31 `compare_tuple< std::u32string >()`

```
template<>
int compare_tuple< std::u32string > (
    const std::vector< std::u32string > & cpp_vector,
    PyObject * op )
```

9.17.2.32 `compare_tuple< std::vector< char > >()`

```
template<>
int compare_tuple< std::vector< char > > (
    const std::vector< std::vector< char >> & cpp_vector,
    PyObject * op )
```

9.17.2.33 `compare_tuple_or_list()`

```
template<typename T , PyObject (*)(const T &) Convert_T_To_Py, T(*) (PyObject *) Convert_Py_To_T,
int(*) (PyObject *) PyUnaryContainer_Check, Py_ssize_t(*) (PyObject *) PyUnaryContainer_Size,
PyObject (*)(PyObject *, size_t) PyUnaryContainer_Get>
int compare_tuple_or_list (
    std::vector< T > const & cpp_vector,
    PyObject * op )
```

Compares a Python tuple or list with a C++ `std::vector`.

Template Parameters

<i>T</i>	C++ type of objects in the vector.
<i>Convert_T_To_Py</i>	Pointer to function to convert a C++ type T to a PyObject*.
<i>Convert_Py_To_T</i>	Pointer to function to convert a PyObject* to a C++ type T.
<i>PyUnaryContainer_Check</i>	A function that takes a PyObject* and returns 1 if it is of a suitable container, 0 otherwise.
<i>PyUnaryContainer_Size</i>	A function that returns the length of the Python container.
<i>PyUnaryContainer_GetA</i>	function that gets a PyObject* from the Python container at a given index as a size_t.

Parameters

<i>cpp_vector</i>	The C++ <code>std::vector</code> .
<i>op</i>	The Python tuple or list.

Returns

0 if identical, non-zero if not.

9.17.2.34 new_py_dict_bytes()

```
PyObject* new_py_dict_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python dict of bytes for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each bytes object to be the key and value. Each key and value will be unique.

Returns

New reference to a dict or NULL on failure.

9.17.2.35 new_py_dict_string()

```
PyObject* new_py_dict_string (
    size_t size,
    size_t str_len )
```

Create a new Python dict of str for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each str object to be the key and value. Each key and value will be unique.

Returns

New reference to a dict or NULL on failure.

9.17.2.36 new_py_dict_string16()

```
PyObject* new_py_dict_string16 (
    size_t size,
    size_t str_len )
```

Create a new Python dict of str (16bit) for both the key and the value.

Parameters

<i>size</i>	Length of the dict.
<i>str_len</i>	Length of each str object to be the key and value. Each key and value will be unique.

Returns

New reference to a `dict` or `NULL` on failure.

9.17.2.37 new_py_dict_string32()

```
PyObject* new_py_dict_string32 (
    size_t size,
    size_t str_len )
```

Create a new Python `dict` of `str` (32bit) for both the key and the value.

Parameters

<i>size</i>	Length of the <code>dict</code> .
<i>str_len</i>	Length of each <code>str</code> object to be the key and value. Each key and value will be unique.

Returns

New reference to a `dict` or `NULL` on failure.

9.17.2.38 new_py_list_bytes()

```
PyObject* new_py_list_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `bytes`.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>bytes</code> object. Each byte is just <code>''</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.17.2.39 new_py_list_string()

```
PyObject* new_py_list_string (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `str`.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each byte is just <code>'</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.17.2.40 `new_py_list_string16()`

```
PyObject* new_py_list_string16 (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `str` with 16 bit characters.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just <code>u'</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.17.2.41 `new_py_list_string32()`

```
PyObject* new_py_list_string32 (
    size_t size,
    size_t str_len )
```

Create a new Python `list` of `str` with 32 bit characters.

Parameters

<i>size</i>	Length of the <code>list</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just <code>U'</code> .

Returns

New reference to a `list` or `NULL` on failure.

9.17.2.42 new_py_set_bytes()

```
PyObject* new_py_set_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python set of bytes.

Parameters

<i>size</i>	Length of the set.
<i>str_len</i>	Length of each bytes object. Each object will be unique.

Returns

New reference to a set or NULL on failure.

9.17.2.43 new_py_set_string()

```
PyObject* new_py_set_string (
    size_t size,
    size_t str_len )
```

Create a new Python set of str.

Parameters

<i>size</i>	Length of the set.
<i>str_len</i>	Length of each str object. Each object will be unique.

Returns

New reference to a set or NULL on failure.

9.17.2.44 new_py_set_u16string()

```
PyObject* new_py_set_u16string (
    size_t size,
    size_t str_len )
```

Create a new Python set of str 16 bit.

Parameters

<i>size</i>	Length of the set.
<i>str_len</i>	Length of each str object. Each object will be unique.

Returns

New reference to a `set` or `NULL` on failure.

9.17.2.45 new_py_set_u32string()

```
PyObject* new_py_set_u32string (
    size_t size,
    size_t str_len )
```

Create a new Python `set` of `str` 32 bit.

Parameters

<i>size</i>	Length of the <code>set</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each object will be unique.

Returns

New reference to a `set` or `NULL` on failure.

9.17.2.46 new_py_tuple_bytes()

```
PyObject* new_py_tuple_bytes (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of `bytes`.

Parameters

<i>size</i>	Length of the <code>tuple</code> .
<i>str_len</i>	Length of each <code>bytes</code> object. Each byte is just <code>'</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.17.2.47 new_py_tuple_string()

```
PyObject* new_py_tuple_string (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of `str`.

Parameters

<i>size</i>	Length of the <code>tuple</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each byte is just <code>'</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.17.2.48 new_py_tuple_string16()

```
PyObject* new_py_tuple_string16 (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of 16 bit unicode `str`.

Parameters

<i>size</i>	Length of the <code>tuple</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just <code>u'</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.17.2.49 new_py_tuple_string32()

```
PyObject* new_py_tuple_string32 (
    size_t size,
    size_t str_len )
```

Create a new Python `tuple` of 32 bit unicode `str`.

Parameters

<i>size</i>	Length of the <code>tuple</code> .
<i>str_len</i>	Length of each <code>str</code> object. Each character is just <code>U'</code> .

Returns

New reference to a `tuple` or `NULL` on failure.

9.17.2.50 test_cpp_std_map_like_to_py_dict()

```
template<template< typename ... > class MapLike, typename K , typename V , PyObject *(*)(const
K &) Convert_K, PyObject *(*)(const V &) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObj←
ect *) Convert_Py_Val>
int test_cpp_std_map_like_to_py_dict (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.51 test_cpp_std_map_to_py_dict()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObject *) Convert_Py_Val>
int test_cpp_std_map_to_py_dict (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.52 test_cpp_std_map_to_py_dict_bytes()

```
int test_cpp_std_map_to_py_dict_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.53 test_cpp_std_map_to_py_dict_string()

```
int test_cpp_std_map_to_py_dict_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.54 test_cpp_std_map_to_py_dict_string16()

```
int test_cpp_std_map_to_py_dict_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.55 test_cpp_std_map_to_py_dict_string32()

```
int test_cpp_std_map_to_py_dict_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.56 test_cpp_std_unordered_map_to_py_dict()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObject *) Convert_Py_Val>
int test_cpp_std_unordered_map_to_py_dict (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.57 test_cpp_std_unordered_map_to_py_dict_bytes()

```
int test_cpp_std_unordered_map_to_py_dict_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.58 test_cpp_std_unordered_map_to_py_dict_string()

```
int test_cpp_std_unordered_map_to_py_dict_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.59 test_cpp_std_unordered_map_to_py_dict_string16()

```
int test_cpp_std_unordered_map_to_py_dict_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.60 test_cpp_std_unordered_map_to_py_dict_string32()

```
int test_cpp_std_unordered_map_to_py_dict_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.61 test_py_dict_to_cpp_std_map()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObject *) Convert_Py_Val>
int test_py_dict_to_cpp_std_map (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.62 test_py_dict_to_cpp_std_map_bytes()

```
int test_py_dict_to_cpp_std_map_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.63 test_py_dict_to_cpp_std_map_like()

```
template<template< typename ... > class MapLike, typename K , typename V , PyObject *(*)(const
K &) Convert_K, PyObject *(*)(const V &) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (Py↵
Object *) Convert_Py_Val>
int test_py_dict_to_cpp_std_map_like (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.64 test_py_dict_to_cpp_std_map_string()

```
int test_py_dict_to_cpp_std_map_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.65 test_py_dict_to_cpp_std_map_string16()

```
int test_py_dict_to_cpp_std_map_string16 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.66 test_py_dict_to_cpp_std_map_string32()

```
int test_py_dict_to_cpp_std_map_string32 (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.67 test_py_dict_to_cpp_std_unordered_map()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V, K(*) (PyObject *) Convert_Py_Key, V(*) (PyObject *) Convert_Py_Val>
int test_py_dict_to_cpp_std_unordered_map (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.68 test_py_dict_to_cpp_std_unordered_map_bytes()

```
int test_py_dict_to_cpp_std_unordered_map_bytes (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.69 test_py_dict_to_cpp_std_unordered_map_string()

```
int test_py_dict_to_cpp_std_unordered_map_string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```


9.17.2.70 test_py_dict_to_cpp_std_unordered_map_u16string()

```
int test_py_dict_to_cpp_std_unordered_map_u16string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.71 test_py_dict_to_cpp_std_unordered_map_u32string()

```
int test_py_dict_to_cpp_std_unordered_map_u32string (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.72 test_py_list_bytes_to_vector()

```
int test_py_list_bytes_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python list to a C++ `std::vector<std::vector<char>>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.73 test_py_list_str16_to_vector()

```
int test_py_list_str16_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python list to a C++ `std::vector<std::u16string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.74 test_py_list_str32_to_vector()

```
int test_py_list_str32_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python `list` to a C++ `std::vector<std::u32string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.75 test_py_list_str_to_vector()

```
int test_py_list_str_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python `list` to a C++ `std::vector<std::string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.76 test_py_list_to_vector()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy, T(*) (PyObject *) ConvertPyToCpp>
int test_py_list_to_vector (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.77 test_py_list_to_vector_round_trip()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_list_to_vector_round_trip (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.78 test_py_set_bytes_to_unordered_set()

```
int test_py_set_bytes_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.79 test_py_set_string16_to_unordered_set()

```
int test_py_set_string16_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.80 test_py_set_string32_to_unordered_set()

```
int test_py_set_string32_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.81 test_py_set_string_to_unordered_set()

```
int test_py_set_string_to_unordered_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.82 test_py_set_to_unordered_set()

```
template<typename T , T(*) (PyObject *) ConvertPyToCpp, PyObject *(*)(const T &) ConvertCppTo←
Py>
int test_py_set_to_unordered_set (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.83 test_py_tuple_bytes_to_vector()

```
int test_py_tuple_bytes_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of bytes to a C++ `std::vector<std::vector<char>>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.84 test_py_tuple_str16_to_vector()

```
int test_py_tuple_str16_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of str to a C++ `std::vector<std::string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.85 test_py_tuple_str32_to_vector()

```
int test_py_tuple_str32_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of str to a C++ `std::vector<std::u32string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.86 test_py_tuple_str_to_vector()

```
int test_py_tuple_str_to_vector (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a Python tuple of str to a C++ `std::vector<std::string>`.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.87 test_py_tuple_to_vector()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_tuple_to_vector (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

Tests a C++ `std::vector` to a Python tuple.

Template Parameters

<i>T</i>	Type of the vector objects.
<i>ConvertCppToPy</i>	Function to convert a C++ <code><T></code> to a Python object.

Parameters

<i>test_results</i>	The test results to update.
<i>type</i>	Type of <code><T></code>
<i>size</i>	Size of the <code>std::vector</code> to create.

Returns

0 on success. Non-zero on failure.

9.17.2.88 test_py_tuple_to_vector_round_trip()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_tuple_to_vector_round_trip (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

Tests a Python tuple to a C++ `std::vector` and back to a Python tuple.

Template Parameters

<i>T</i>	Type of the vector objects.
<i>ConvertCppToPy</i>	Function to convert a C++ <code><T></code> to a Python object.

Parameters

<i>test_results</i>	The test results to update.
<i>type</i>	Type of <T>
<i>size</i>	Size of the <code>std::vector</code> to create.

Returns

0 on success. Non-zero on failure.

9.17.2.89 test_unordered_set_bytes_to_py_set()

```
int test_unordered_set_bytes_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.90 test_unordered_set_string_to_py_set()

```
int test_unordered_set_string_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.91 test_unordered_set_to_py_set()

```
template<typename T , T(*) (PyObject *) ConvertPyToCpp, PyObject *(*)(const T &) ConvertCppToPy>
int test_unordered_set_to_py_set (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.92 test_unordered_set_u16string_to_py_set()

```
int test_unordered_set_u16string_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.93 test_unordered_set_u32string_to_py_set()

```
int test_unordered_set_u32string_to_py_set (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

9.17.2.94 test_vector_string_to_py_list()

```
int test_vector_string_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::string>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.95 test_vector_string_to_py_tuple()

```
int test_vector_string_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::string>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.96 test_vector_to_py_list()

```
template<typename T >
int test_vector_to_py_list (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

Tests a Python vector to a C++ std::vector.

Template Parameters

<i>T</i>	Type of the vector objects.
----------	-----------------------------

Parameters

<i>test_results</i>	The test results to update.
<i>type</i>	Type of <T>
<i>size</i>	Size of the std::vector to create.

Returns

0 on success. Non-zero on failure.

9.17.2.97 test_vector_to_py_list_round_trip()

```
template<typename T >
int test_vector_to_py_list_round_trip (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.17.2.98 test_vector_to_py_tuple()

```
template<typename T , T(*) (PyObject *) ConvertPyToCpp>
int test_vector_to_py_tuple (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

Tests a C++ std::vector to a Python tuple.

Template Parameters

<i>T</i>	Type of the vector objects.
<i>ConvertPyToCpp</i>	Function to convert a Python object to a C++ <T>

Parameters

<i>test_results</i>	The test results to update.
<i>type</i>	Type of <T>
<i>size</i>	Size of the <code>std::vector</code> to create.

Returns

0 on success. Non-zero on failure.

9.17.2.99 test_vector_to_py_tuple_round_trip()

```
template<typename T >
int test_vector_to_py_tuple_round_trip (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

Tests a C++ `std::vector` to a Python tuple and back to a C++ `std::vector`.

Template Parameters

<i>T</i>	Type of the vector objects.
----------	-----------------------------

Parameters

<i>test_results</i>	The test results to update.
<i>type</i>	Type of <T>
<i>size</i>	Size of the <code>std::vector</code> to create.

Returns

0 on success. Non-zero on failure.

9.17.2.100 test_vector_u16string_to_py_list()

```
int test_vector_u16string_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u16string>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.101 test_vector_u16string_to_py_tuple()

```
int test_vector_u16string_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::string>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.102 test_vector_u32string_to_py_list()

```
int test_vector_u32string_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u32string>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.103 test_vector_u32string_to_py_tuple()

```
int test_vector_u32string_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::u32string>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.104 test_vector_vector_char_to_py_list()

```
int test_vector_vector_char_to_py_list (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::vector<char>>` to a Python list.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.2.105 test_vector_vector_char_to_py_tuple()

```
int test_vector_vector_char_to_py_tuple (
    TestResults & test_results,
    size_t size,
    size_t str_len )
```

Tests converting a C++ `std::vector<std::vector<char>>` to a Python tuple.

Parameters

<i>test_results</i>	The test results to update.
<i>size</i>	Size of the <code>std::vector</code> to create.
<i>str_len</i>	Length of each entry in the <code>std::vector</code> .

Returns

0 on success, non-zero on failure.

9.17.3 Variable Documentation

9.17.3.1 PY_ERR_ON_ENTRY_RETURN_CODE

```
const int PY_ERR_ON_ENTRY_RETURN_CODE = -1
```

9.17.3.2 PY_ERR_ON_EXIT_RETURN_CODE

```
const int PY_ERR_ON_EXIT_RETURN_CODE = -2
```

9.18 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.cpp File Reference

```
#include <Python.h>
#include "cpy/python_convert.h"
#include "test_functional.h"
```

Functions

- void [test_example_vector_to_py_tuple_double](#) ()
- void [test_example_py_tuple_to_vector_double](#) ()
- void [test_example_cpp_std_unordered_map_to_py_dict](#) ()
- void [test_example_cpp_std_map_to_py_dict](#) ()
- void [test_example_py_dict_to_cpp_std_unordered_map](#) ()
- void [test_functional_tuple_setitem](#) (TestResultS &test_results)
- void [test_functional_list_setitem](#) (TestResultS &test_results)
- void [test_functional_set_add](#) (TestResultS &test_results)
- void [test_functional_set_add_from_iterable](#) (TestResultS &test_results)
- void [test_functional_frozenset_add](#) (TestResultS &test_results)
- void [test_functional_frozenset_add_from_iterable](#) (TestResultS &test_results)
- void [test_functional_dict_setitem](#) (TestResultS &test_results)
- void [test_functional_dict_copy](#) (TestResultS &test_results)
- void [test_functional_tuple](#) (TestResultS &test_results)
- void [test_functional_list](#) (TestResultS &test_results)
- void [test_functional_set](#) (TestResultS &test_results)
- void [test_functional_dict_with_std_unordred_map](#) (TestResultS &test_results)
- void [test_functional_dict_with_std_map](#) (TestResultS &test_results)
- void [test_functional_all](#) (TestResultS &test_results)

9.18.1 Function Documentation

9.18.1.1 test_example_cpp_std_map_to_py_dict()

```
void test_example_cpp_std_map_to_py_dict ( )
```

Demonstration code to convert a `std::map<long, std::string>` to a Python dict.

9.18.1.2 test_example_cpp_std_unordered_map_to_py_dict()

```
void test_example_cpp_std_unordered_map_to_py_dict ( )
```

Demonstration code to convert a `std::unordered_map<long, std::string>` to a Python dict.

9.18.1.3 test_example_py_dict_to_cpp_std_unordered_map()

```
void test_example_py_dict_to_cpp_std_unordered_map ( )
```

Demonstration code to convert a Python dict[int, bytes] to a `std::unordered_map<long, std::string>`.

9.18.1.4 test_example_py_tuple_to_vector_double()

```
void test_example_py_tuple_to_vector_double ( )
```

Demonstration code for converting a Python tuple off floats to a `std::vector<double>`.

9.18.1.5 test_example_vector_to_py_tuple_double()

```
void test_example_vector_to_py_tuple_double ( )
```

Demonstration code for converting a `std::vector<double>` to a Python tuple.

9.18.1.6 test_functional_all()

```
void test_functional_all (
    TestResults & test_results )
```

9.18.1.7 test_functional_dict_copy()

```
void test_functional_dict_copy (
    TestResults & test_results )
```

Tests the reference count changes when using `PyDict_Copy()`

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.8 test_functional_dict_setitem()

```
void test_functional_dict_setitem (
    TestResults & test_results )
```

Tests the reference count changes when inserting into a dict with `PyDict_SetItem()`

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.9 test_functional_dict_with_std_map()

```
void test_functional_dict_with_std_map (
    TestResults & test_results )
```

9.18.1.10 test_functional_dict_with_std_unordred_map()

```
void test_functional_dict_with_std_unordred_map (  
    TestResults & test_results )
```

9.18.1.11 test_functional_frozenset_add()

```
void test_functional_frozenset_add (  
    TestResults & test_results )
```

Tests the reference count changes when inserting into a frozenset with PySet_Add()

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.12 test_functional_frozenset_add_from_iterable()

```
void test_functional_frozenset_add_from_iterable (  
    TestResults & test_results )
```

Tests the reference count changes when inserting into a frozenset with PySet_Add() No error checking.

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.13 test_functional_list()

```
void test_functional_list (  
    TestResults & test_results )
```

9.18.1.14 test_functional_list_setitem()

```
void test_functional_list_setitem (  
    TestResults & test_results )
```

Tests the reference count changes when inserting into a list with PyList_SetItem()

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.15 test_functional_set()

```
void test_functional_set (
    TestResults & test_results )
```

9.18.1.16 test_functional_set_add()

```
void test_functional_set_add (
    TestResults & test_results )
```

Tests the reference count changes when inserting into a set with PySet_Add()

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.17 test_functional_set_add_from_iterable()

```
void test_functional_set_add_from_iterable (
    TestResults & test_results )
```

Tests the reference count changes when inserting into a set with PySet_New(iterable) No error checking.

Parameters

<i>test_results</i>	
---------------------	--

9.18.1.18 test_functional_tuple()

```
void test_functional_tuple (
    TestResults & test_results )
```

9.18.1.19 test_functional_tuple_setitem()

```
void test_functional_tuple_setitem (
    TestResultS & test_results )
```

Tests the reference count changes when inserting into a tuple with PyTuple_SetItem()

Parameters

<i>test_results</i>	
---------------------	--

9.19 /Users/paulross/CLionProjects/PythonCppHomogeneous↔ Containers/src/cpy/tests/test_functional.h File Reference

```
#include "test_common.h"
```

Functions

- void [test_functional_all](#) (TestResultS &test_results)

9.19.1 Function Documentation

9.19.1.1 test_functional_all()

```
void test_functional_all (
    TestResultS & test_results )
```

9.20 /Users/paulross/CLionProjects/PythonCppHomogeneous↔ Containers/src/cpy/tests/test_internal.cpp File Reference

```
#include "test_internal.h"
```

Functions

- int `doubles_cmp` (double a, double b)
- void `test_internal_test_result_test_count` (TestResultS &test_results)
- void `test_internal_test_result_total_time` (TestResultS &test_results)
- void `test_internal_test_result_exec_time_min_max` (TestResultS &test_results)
- void `test_internal_test_result_exec_time` (TestResultS &test_results)
- void `test_internal_test_result_string` (TestResultS &test_results)
- void `test_internal_test_result_string_multiple_a` (TestResultS &test_results)
- void `test_internal_test_result_string_multiple_b` (TestResultS &test_results)
- void `test_internal_test_result_string_using_rate` (TestResultS &test_results)
- void `test_internal_test_result_atomic_test_mean_exec_time` (TestResultS &test_results)
- void `test_internal_all` (TestResultS &test_results)

9.20.1 Function Documentation

9.20.1.1 `doubles_cmp()`

```
int doubles_cmp (
    double a,
    double b )
```

Compare two doubles to see if they are 'similar'. If possible this normalises the two values to unity before comparing them with `std::numeric_limits<double>::epsilon()`

Parameters

<i>a</i>	
<i>b</i>	

Returns

Zero if the two arguments are equal, non-zero if they are out of range. This allows: `result |= doubles←
_cmp(value, expected);`

9.20.1.2 `test_internal_all()`

```
void test_internal_all (
    TestResultS & test_results )
```

9.20.1.3 `test_internal_test_result_atomic_test_mean_exec_time()`

```
void test_internal_test_result_atomic_test_mean_exec_time (
    TestResultS & test_results )
```

9.20.1.4 test_internal_test_result_exec_time()

```
void test_internal_test_result_exec_time (
    TestResults & test_results )
```

9.20.1.5 test_internal_test_result_exec_time_min_max()

```
void test_internal_test_result_exec_time_min_max (
    TestResults & test_results )
```

9.20.1.6 test_internal_test_result_string()

```
void test_internal_test_result_string (
    TestResults & test_results )
```

9.20.1.7 test_internal_test_result_string_multiple_a()

```
void test_internal_test_result_string_multiple_a (
    TestResults & test_results )
```

9.20.1.8 test_internal_test_result_string_multiple_b()

```
void test_internal_test_result_string_multiple_b (
    TestResults & test_results )
```

9.20.1.9 test_internal_test_result_string_using_rate()

```
void test_internal_test_result_string_using_rate (
    TestResults & test_results )
```

Example of a very fast test where using the "Rate" column can recover the mean time with accuracy.

Parameters

<i>test_results</i>	
---------------------	--

9.20.1.10 test_internal_test_result_test_count()

```
void test_internal_test_result_test_count (
    TestResults & test_results )
```

This demonstrates that the number of tests is the sum of all the test_counts for each added result.

Parameters

<i>test_results</i>	
---------------------	--

9.20.1.11 test_internal_test_result_total_time()

```
void test_internal_test_result_total_time (
    TestResults & test_results )
```

This demonstrates that the total execution time is the sum of all the sub-test execution time.

Parameters

<i>test_results</i>	
---------------------	--

9.21 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.h File Reference

```
#include "test_common.h"
```

Functions

- void [test_internal_all](#) (TestResults &test_results)

9.21.1 Function Documentation

9.21.1.1 test_internal_all()

```
void test_internal_all (
    TestResults & test_results )
```

9.22 /Users/paulross/CLionProjects/PythonCppHomogeneous Containers/src/cpy/tests/test_memory.cpp File Reference

```
#include "test_memory.h"
#include <Python.h>
#include "cpp/get_rss.h"
#include "cpy/python_convert.h"
#include "test_common.h"
```

Functions

- int [test_memory_py_tuple_float](#) (TestResultS &test_results, const std::string &type, size_t size)
- int [test_memory_py_dict](#) (TestResultS &test_results, const std::string &type, size_t size)
- int [test_memory_vector_vector_char_to_py_tuple](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_vector_vector_char_to_py_set](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_vector_vector_char_to_py_set_special](#) (TestResultS &test_results)
- int [test_memory_py_tuple_vector_char_to_vector](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_test_vector_string_to_py_tuple](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_py_tuple_unicode8_to_vector](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_vector_u16string_to_py_tuple](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_py_tuple_str16_to_vector](#) (TestResultS &test_results, size_t str_len_min, size_t str_len_↵_max, size_t size_min, size_t size_max)
- int [test_memory_vector_u32string_to_py_tuple](#) (TestResultS &test_results, size_t str_len_min, size_t str_↵_len_max, size_t size_min, size_t size_max)
- int [test_memory_py_tuple_str32_to_vector](#) (TestResultS &test_results, size_t str_len_min, size_t str_len_↵_max, size_t size_min, size_t size_max)
- void [test_memory_all](#) (TestResultS &test_results)

9.22.1 Function Documentation

9.22.1.1 test_memory_all()

```
void test_memory_all (
    TestResultS & test_results )
```

9.22.1.2 test_memory_py_dict()

```
int test_memory_py_dict (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.22.1.3 test_memory_py_tuple_float()

```
int test_memory_py_tuple_float (
    TestResults & test_results,
    const std::string & type,
    size_t size )
```

9.22.1.4 test_memory_py_tuple_str16_to_vector()

```
int test_memory_py_tuple_str16_to_vector (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

Tests the memory usage when converting a Python tuple of 16 bit strings to a C++ vector of std::u16string.

9.22.1.5 test_memory_py_tuple_str32_to_vector()

```
int test_memory_py_tuple_str32_to_vector (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

Tests the memory usage when converting a Python tuple of 16 bit strings to a C++ vector of std::u16string.

9.22.1.6 test_memory_py_tuple_unicode8_to_vector()

```
int test_memory_py_tuple_unicode8_to_vector (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

Tests the memory usage when converting a Python tuple of 8 bit strings to a C++ vector of std::string.

9.22.1.7 test_memory_py_tuple_vector_char_to_vector()

```
int test_memory_py_tuple_vector_char_to_vector (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

9.22.1.8 test_memory_test_vector_string_to_py_tuple()

```
int test_memory_test_vector_string_to_py_tuple (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

Tests the memory usage when converting a C++ vector of std::string to a Python to a tuple of 8 bit strings.

9.22.1.9 test_memory_vector_u16string_to_py_tuple()

```
int test_memory_vector_u16string_to_py_tuple (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

Tests the memory usage when converting a C++ vector of std::u16string to a Python to a tuple of 16 bit strings.

9.22.1.10 test_memory_vector_u32string_to_py_tuple()

```
int test_memory_vector_u32string_to_py_tuple (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

Tests the memory usage when converting a C++ vector of std::u16string to a Python to a tuple of 16 bit strings.

9.22.1.11 test_memory_vector_vector_char_to_py_set()

```
int test_memory_vector_vector_char_to_py_set (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```


9.22.1.12 test_memory_vector_vector_char_to_py_set_special()

```
int test_memory_vector_vector_char_to_py_set_special (
    TestResults & test_results )
```

Special test for memory leaks from set conversion. Set of 1024 strings of 1Mb length

Parameters

<i>test_results</i>	
---------------------	--

Returns

9.22.1.13 test_memory_vector_vector_char_to_py_tuple()

```
int test_memory_vector_vector_char_to_py_tuple (
    TestResults & test_results,
    size_t str_len_min,
    size_t str_len_max,
    size_t size_min,
    size_t size_max )
```

9.23 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.h File Reference

```
#include "test_common.h"
```

Functions

- void [test_memory_all](#) (TestResults &test_results)

9.23.1 Function Documentation

9.23.1.1 test_memory_all()

```
void test_memory_all (
    TestResults & test_results )
```

9.24 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.cpp File Reference

```
#include <Python.h>
#include "cpy/python_convert.h"
#include "test_common.h"
#include "test_performance.h"
```

Macros

- #define TEST_PERFORMANCE_FUNDAMENTAL_TYPES
- #define TEST_PERFORMANCE_OBJECT_BOOL
- #define TEST_PERFORMANCE_OBJECT_LONG
- #define TEST_PERFORMANCE_OBJECT_DOUBLE
- #define TEST_PERFORMANCE_OBJECT_COMPLEX
- #define TEST_PERFORMANCE_OBJECT_BYTES
- #define TEST_PERFORMANCE_OBJECT_STRING
- #define TEST_PERFORMANCE_OBJECT_STRING_16
- #define TEST_PERFORMANCE_OBJECT_STRING_32
- #define TEST_PERFORMANCE_TUPLES
- #define TEST_PERFORMANCE_LISTS
- #define TEST_PERFORMANCE_SETS
- #define TEST_PERFORMANCE_DICTS

Functions

- int test_bool_to_py_bool_multiple (TestResultS &test_results, size_t test_count, size_t repeat)
- int test_py_bool_to_cpp_bool_multiple (TestResultS &test_results, size_t test_count, size_t repeat)
- int test_long_to_py_int_multiple (TestResultS &test_results, size_t size, size_t repeat)
- int test_py_int_to_cpp_long_multiple (TestResultS &test_results, size_t size, size_t repeat)
- int test_double_to_py_float_multiple (TestResultS &test_results, size_t size, size_t repeat)
- int test_py_float_to_cpp_double_multiple (TestResultS &test_results, size_t size, size_t repeat)
- int test_complex_to_py_complex_multiple (TestResultS &test_results, size_t size, size_t repeat)
- int test_py_complex_to_cpp_complex_multiple (TestResultS &test_results, size_t size, size_t repeat)
- int test_cpp_vector_char_to_py_bytes_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_py_bytes_to_cpp_vector_char_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_cpp_string_to_py_str_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_py_str_to_cpp_string_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_cpp_u16string_to_py_str16_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_py_str16_to_cpp_u16string_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_cpp_u32string_to_py_str32_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)
- int test_py_str32_to_cpp_u32string_multiple (TestResultS &test_results, size_t string_size, size_t test_count, size_t repeat)

- `template<template< typename ... > class ListLike, typename T >`
`int test_list_like_to_py_tuple_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat, const std::string &container_type)`
- `template<typename T >`
`int test_vector_to_py_tuple_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T >`
`int test_list_to_py_tuple_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T >`
`int test_perf_vector_to_py_tuple_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename T >`
`int test_perf_list_to_py_tuple_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<template< typename ... > class ListLike, typename T , PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_tuple_to_list_like_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat, const std::string &container_type)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_tuple_to_vector_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_tuple_to_list_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`int test_perf_py_tuple_to_vector_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename T , PyObject (*)(const T &) ConvertCppToPy>`
`int test_perf_py_tuple_to_list_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_list_like_vector_char_to_py_tuple_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_vector_vector_char_to_py_tuple_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_list_vector_char_to_py_tuple_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_vector_vector_char_to_py_tuple_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_list_vector_char_to_py_tuple_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_py_tuple_bytes_to_list_like_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_py_tuple_bytes_to_vector_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_tuple_bytes_to_list_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_tuple_to_vector_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_tuple_to_list_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_list_like_string_to_py_tuple_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_vector_string_to_py_tuple_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_list_string_to_py_tuple_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_vector_string_to_py_tuple_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_list_string_to_py_tuple_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_py_tuple_str_to_list_like_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_py_tuple_str_to_vector_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`

- `int test_py_tuple_str_to_list_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_tuple_to_vector_string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_tuple_to_list_string_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike, typename T >`
`int test_list_like_to_py_list_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat, const std::string &container_type)`
- `template<typename T >`
`int test_vector_to_py_list_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T >`
`int test_list_to_py_list_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T >`
`int test_perf_vector_to_py_list_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename T >`
`int test_perf_list_to_py_list_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<template< typename ... > class ListLike, typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_list_to_list_like_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat, const std::string &container_type)`
- `template<typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_list_to_vector_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_list_to_list_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_perf_py_list_to_vector_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_perf_py_list_to_list_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_list_like_vector_char_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_vector_vector_char_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_list_vector_char_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_vector_vector_char_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_list_vector_char_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_py_list_bytes_to_list_like_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_py_list_bytes_to_vector_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_list_bytes_to_list_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_list_to_vector_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_list_to_list_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_list_like_string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `template<template< typename ... > class ListLike>`
`int test_list_like_u16string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `template<template< typename ... > class ListLike>`
`int test_list_like_u32string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_vector_string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`

- `int test_list_string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_vector_string_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_list_string_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `int test_vector_u16string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_list_u16string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_vector_u16string_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_list_u16string_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `int test_vector_u32string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_list_u32string_to_py_list_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_vector_u32string_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_list_u32string_to_py_list_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class ListLike>`
`int test_py_list_str_to_list_like_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `template<template< typename ... > class ListLike>`
`int test_py_list_str16_to_list_like_u16string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `template<template< typename ... > class ListLike>`
`int test_py_list_str32_to_list_like_u32string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_py_list_str_to_vector_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_list_str_to_list_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_list_to_vector_string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_list_to_list_string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_py_list_str16_to_vector_u16string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_list_str16_to_list_u16string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_list_to_vector_u16string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_list_to_list_u16string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_py_list_str32_to_vector_u32string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_list_str32_to_list_u32string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_list_to_vector_u32string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_list_to_list_u32string_multiple (TestResultS &test_results, size_t repeat)`
- `template<typename T >`
`int test_unordered_set_to_py_set_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T >`
`int test_perf_unordered_set_to_py_set_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_py_set_to_unordered_set_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename T, PyObject (*)(const T &) ConvertCppToPy>`
`int test_perf_py_set_to_unordered_set_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `int test_unordered_set_vector_char_to_py_set_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_unordered_set_vector_char_to_py_set_multiple (TestResultS &test_results, size_t repeat)`
- `int test_py_set_bytes_to_unordered_set_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`

- `int test_perf_py_set_bytes_to_unordered_set_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `int test_unordered_set_string_to_py_set_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_unordered_set_string_to_py_set_multiple (TestResultS &test_results, size_t repeat)`
- `int test_py_set_str_to_unordered_set_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_set_str_to_unordered_set_string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_unordered_set_u16string_to_py_set_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_unordered_set_u16string_to_py_set_multiple (TestResultS &test_results, size_t repeat)`
- `int test_py_set_str16_to_unordered_set_u16string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_set_str16_to_unordered_set_u16string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_unordered_set_u32string_to_py_set_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_unordered_set_u32string_to_py_set_multiple (TestResultS &test_results, size_t repeat)`
- `int test_py_set_str32_to_unordered_set_u32string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_set_str32_to_unordered_set_u32string_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class MapLike, typename K , typename V >
int test_cpp_std_map_like_to_py_dict_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat, const std::string &container_type)`
- `template<typename K , typename V >
int test_cpp_std_unordered_map_to_py_dict_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename K , typename V >
int test_cpp_std_map_to_py_dict_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename K , typename V >
int test_perf_cpp_std_unordered_map_to_py_dict_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename K , typename V >
int test_perf_cpp_std_map_to_py_dict_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<template< typename ... > class MapLike, typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>
int test_py_dict_to_cpp_std_map_like_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat, const std::string &container_type)`
- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>
int test_py_dict_to_cpp_std_unordered_map_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>
int test_py_dict_to_cpp_std_map_multiple (TestResultS &test_results, const std::string &type, size_t size, size_t repeat)`
- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>
int test_perf_py_dict_to_cpp_std_unordered_map_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<typename K , typename V , PyObject (*)(const K &) Convert_K, PyObject (*)(const V &) Convert_V>
int test_perf_py_dict_to_cpp_std_map_multiple (TestResultS &test_results, const std::string &type, size_t repeat)`
- `template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_cpp_std_unordered_map_to_py_dict_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`

- `int test_cpp_std_map_to_py_dict_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_cpp_std_unordered_map_to_py_dict_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_cpp_std_map_to_py_dict_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_py_dict_to_cpp_std_unordered_map_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_dict_to_cpp_std_map_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_dict_to_cpp_std_unordered_map_vector_char_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_dict_to_cpp_std_map_vector_char_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_cpp_std_unordered_map_to_py_dict_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_cpp_std_map_to_py_dict_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_cpp_std_unordered_map_to_py_dict_string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_cpp_std_map_to_py_dict_string_multiple (TestResultS &test_results, size_t repeat)`
- `template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat, const std::string &container_type)`
- `int test_py_dict_to_cpp_std_unordered_map_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_py_dict_to_cpp_std_map_string_multiple (TestResultS &test_results, size_t size, size_t str_len, size_t repeat)`
- `int test_perf_py_dict_to_cpp_std_unordered_map_string_multiple (TestResultS &test_results, size_t repeat)`
- `int test_perf_py_dict_to_cpp_std_map_string_multiple (TestResultS &test_results, size_t repeat)`
- `void test_performance_all (TestResultS &test_results)`

Variables

- `const size_t MIN_SIZE_OF_CONTAINER = 1`
- `const size_t LIMIT_SIZE_OF_CONTAINER = 1 << 21`
- `const size_t LIMIT_SIZE_OF_CONTAINER_DICT = 1 << 21`
- `const size_t INC_SIZE_OF_CONTAINER_MULTIPLE = 2`
- `const size_t TEST_REPEAT = 5`
- `const size_t MIN_STRING_LENGTH_NON_HASHABLE = 2`
- `const size_t MIN_STRING_LENGTH_HASHABLE = 16`
- `const size_t LIMIT_STRING_LENGTH = 1024 * 2`
- `const size_t INC_STRING_LENGTH_MULTIPLE = 8`

9.24.1 Macro Definition Documentation

9.24.1.1 TEST_PERFORMANCE_DICTS

```
#define TEST_PERFORMANCE_DICTS
```

9.24.1.2 TEST_PERFORMANCE_FUNDAMENTAL_TYPES

```
#define TEST_PERFORMANCE_FUNDAMENTAL_TYPES
```

9.24.1.3 TEST_PERFORMANCE_LISTS

```
#define TEST_PERFORMANCE_LISTS
```

9.24.1.4 TEST_PERFORMANCE_OBJECT_BOOL

```
#define TEST_PERFORMANCE_OBJECT_BOOL
```

9.24.1.5 TEST_PERFORMANCE_OBJECT_BYTES

```
#define TEST_PERFORMANCE_OBJECT_BYTES
```

9.24.1.6 TEST_PERFORMANCE_OBJECT_COMPLEX

```
#define TEST_PERFORMANCE_OBJECT_COMPLEX
```

9.24.1.7 TEST_PERFORMANCE_OBJECT_DOUBLE

```
#define TEST_PERFORMANCE_OBJECT_DOUBLE
```

9.24.1.8 TEST_PERFORMANCE_OBJECT_LONG

```
#define TEST_PERFORMANCE_OBJECT_LONG
```


9.24.1.9 TEST_PERFORMANCE_OBJECT_STRING

```
#define TEST_PERFORMANCE_OBJECT_STRING
```

9.24.1.10 TEST_PERFORMANCE_OBJECT_STRING_16

```
#define TEST_PERFORMANCE_OBJECT_STRING_16
```

9.24.1.11 TEST_PERFORMANCE_OBJECT_STRING_32

```
#define TEST_PERFORMANCE_OBJECT_STRING_32
```

9.24.1.12 TEST_PERFORMANCE_SETS

```
#define TEST_PERFORMANCE_SETS
```

9.24.1.13 TEST_PERFORMANCE_TUPLES

```
#define TEST_PERFORMANCE_TUPLES
```

9.24.2 Function Documentation

9.24.2.1 test_bool_to_py_bool_multiple()

```
int test_bool_to_py_bool_multiple (  
    TestResults & test_results,  
    size_t test_count,  
    size_t repeat )
```

9.24.2.2 test_complex_to_py_complex_multiple()

```
int test_complex_to_py_complex_multiple (
    TestResults & test_results,
    size_t size,
    size_t repeat )
```

9.24.2.3 test_cpp_std_map_like_to_py_dict_multiple()

```
template<template< typename ... > class MapLike, typename K , typename V >
int test_cpp_std_map_like_to_py_dict_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.4 test_cpp_std_map_like_to_py_dict_string_multiple()

```
template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.5 test_cpp_std_map_like_to_py_dict_vector_char_multiple()

```
template<template< typename ... > class MapLike>
int test_cpp_std_map_like_to_py_dict_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.6 test_cpp_std_map_to_py_dict_multiple()

```
template<typename K , typename V >
int test_cpp_std_map_to_py_dict_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.7 test_cpp_std_map_to_py_dict_string_multiple()

```
int test_cpp_std_map_to_py_dict_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.8 test_cpp_std_map_to_py_dict_vector_char_multiple()

```
int test_cpp_std_map_to_py_dict_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.9 test_cpp_std_unordered_map_to_py_dict_multiple()

```
template<typename K , typename V >
int test_cpp_std_unordered_map_to_py_dict_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.10 test_cpp_std_unordered_map_to_py_dict_string_multiple()

```
int test_cpp_std_unordered_map_to_py_dict_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.11 test_cpp_std_unordered_map_to_py_dict_vector_char_multiple()

```
int test_cpp_std_unordered_map_to_py_dict_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.12 test_cpp_string_to_py_str_multiple()

```
int test_cpp_string_to_py_str_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.13 test_cpp_u16string_to_py_str16_multiple()

```
int test_cpp_u16string_to_py_str16_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.14 test_cpp_u32string_to_py_str32_multiple()

```
int test_cpp_u32string_to_py_str32_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.15 test_cpp_vector_char_to_py_bytes_multiple()

```
int test_cpp_vector_char_to_py_bytes_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

Create a C++ `std::string` of the length `string_size` then test converting this to a Python `bytes` this `repeat` times. The timing for each test is made by summing the conversion `test_count` times. A single `TestResult` is added to the `tests_results` which will have `repeat` individual execution timings.

Parameters

<i>test_results</i>	The list of test results to add a single <code>TestResult</code> .
<i>string_size</i>	The length of the string to convert.
<i>test_count</i>	The number of times the conversion is made for the measured execution time. Divide the measured execution time by this number to get the individual conversion execution time. Chose this to be higher for fast conversions, lower (or even unity) for slow ones.
<i>repeat</i>	The number of individual execution timings that make up a single test. Chose this to give a decent min/mean/std. dev./max. About 5 seems about right.

Returns

Zero.

9.24.2.16 test_double_to_py_float_multiple()

```
int test_double_to_py_float_multiple (
    TestResults & test_results,
    size_t size,
    size_t repeat )
```

9.24.2.17 test_list_like_string_to_py_list_multiple()

```
template<template< typename ... > class ListLike>
int test_list_like_string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.18 test_list_like_string_to_py_tuple_multiple()

```
template<template< typename ... > class ListLike>
int test_list_like_string_to_py_tuple_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.19 test_list_like_to_py_list_multiple()

```
template<template< typename ... > class ListLike, typename T >
int test_list_like_to_py_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.20 test_list_like_to_py_tuple_multiple()

```
template<template< typename ... > class ListLike, typename T >
int test_list_like_to_py_tuple_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.21 test_list_like_u16string_to_py_list_multiple()

```
template<template< typename ... > class ListLike>
int test_list_like_u16string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.22 test_list_like_u32string_to_py_list_multiple()

```
template<template< typename ... > class ListLike>
int test_list_like_u32string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.23 test_list_like_vector_char_to_py_list_multiple()

```
template<template< typename ... > class ListLike>
int test_list_like_vector_char_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.24 test_list_like_vector_char_to_py_tuple_multiple()

```
template<template< typename ... > class ListLike>
int test_list_like_vector_char_to_py_tuple_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.25 test_list_string_to_py_list_multiple()

```
int test_list_string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.26 test_list_string_to_py_tuple_multiple()

```
int test_list_string_to_py_tuple_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.27 test_list_to_py_list_multiple()

```
template<typename T >
int test_list_to_py_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.28 test_list_to_py_tuple_multiple()

```
template<typename T >
int test_list_to_py_tuple_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.29 test_list_u16string_to_py_list_multiple()

```
int test_list_u16string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.30 test_list_u32string_to_py_list_multiple()

```
int test_list_u32string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.31 test_list_vector_char_to_py_list_multiple()

```
int test_list_vector_char_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.32 test_list_vector_char_to_py_tuple_multiple()

```
int test_list_vector_char_to_py_tuple_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.33 test_long_to_py_int_multiple()

```
int test_long_to_py_int_multiple (
    TestResults & test_results,
    size_t size,
    size_t repeat )
```


9.24.2.34 test_perf_cpp_std_map_to_py_dict_multiple()

```
template<typename K , typename V >
int test_perf_cpp_std_map_to_py_dict_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.35 test_perf_cpp_std_map_to_py_dict_string_multiple()

```
int test_perf_cpp_std_map_to_py_dict_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.36 test_perf_cpp_std_map_to_py_dict_vector_char_multiple()

```
int test_perf_cpp_std_map_to_py_dict_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.37 test_perf_cpp_std_unordered_map_to_py_dict_multiple()

```
template<typename K , typename V >
int test_perf_cpp_std_unordered_map_to_py_dict_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.38 test_perf_cpp_std_unordered_map_to_py_dict_string_multiple()

```
int test_perf_cpp_std_unordered_map_to_py_dict_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.39 test_perf_cpp_std_unordered_map_to_py_dict_vector_char_multiple()

```
int test_perf_cpp_std_unordered_map_to_py_dict_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.40 test_perf_list_string_to_py_list_multiple()

```
int test_perf_list_string_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.41 test_perf_list_string_to_py_tuple_multiple()

```
int test_perf_list_string_to_py_tuple_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.42 test_perf_list_to_py_list_multiple()

```
template<typename T >
int test_perf_list_to_py_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.43 test_perf_list_to_py_tuple_multiple()

```
template<typename T >
int test_perf_list_to_py_tuple_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.44 test_perf_list_u16string_to_py_list_multiple()

```
int test_perf_list_u16string_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.45 test_perf_list_u32string_to_py_list_multiple()

```
int test_perf_list_u32string_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.46 test_perf_list_vector_char_to_py_list_multiple()

```
int test_perf_list_vector_char_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.47 test_perf_list_vector_char_to_py_tuple_multiple()

```
int test_perf_list_vector_char_to_py_tuple_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.48 test_perf_py_dict_to_cpp_std_map_multiple()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V>
int test_perf_py_dict_to_cpp_std_map_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.49 test_perf_py_dict_to_cpp_std_map_string_multiple()

```
int test_perf_py_dict_to_cpp_std_map_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.50 test_perf_py_dict_to_cpp_std_map_vector_char_multiple()

```
int test_perf_py_dict_to_cpp_std_map_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.51 test_perf_py_dict_to_cpp_std_unordered_map_multiple()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V>
int test_perf_py_dict_to_cpp_std_unordered_map_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.52 test_perf_py_dict_to_cpp_std_unordered_map_string_multiple()

```
int test_perf_py_dict_to_cpp_std_unordered_map_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.53 test_perf_py_dict_to_cpp_std_unordered_map_vector_char_multiple()

```
int test_perf_py_dict_to_cpp_std_unordered_map_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.54 test_perf_py_list_to_list_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_perf_py_list_to_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.55 test_perf_py_list_to_list_string_multiple()

```
int test_perf_py_list_to_list_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.56 test_perf_py_list_to_list_u16string_multiple()

```
int test_perf_py_list_to_list_u16string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.57 test_perf_py_list_to_list_u32string_multiple()

```
int test_perf_py_list_to_list_u32string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.58 test_perf_py_list_to_list_vector_char_multiple()

```
int test_perf_py_list_to_list_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.59 test_perf_py_list_to_vector_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_perf_py_list_to_vector_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.60 test_perf_py_list_to_vector_string_multiple()

```
int test_perf_py_list_to_vector_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.61 test_perf_py_list_to_vector_u16string_multiple()

```
int test_perf_py_list_to_vector_u16string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.62 test_perf_py_list_to_vector_u32string_multiple()

```
int test_perf_py_list_to_vector_u32string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.63 test_perf_py_list_to_vector_vector_char_multiple()

```
int test_perf_py_list_to_vector_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.64 test_perf_py_set_bytes_to_unordered_set_vector_char_multiple()

```
int test_perf_py_set_bytes_to_unordered_set_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.65 test_perf_py_set_str16_to_unordered_set_u16string_multiple()

```
int test_perf_py_set_str16_to_unordered_set_u16string_multiple (
    TestResults & test_results,
    size_t repeat )
```

Invoke `test_py_set_str16_to_unordered_set_u16string_multiple()` with different size values and containers.

9.24.2.66 test_perf_py_set_str32_to_unordered_set_u32string_multiple()

```
int test_perf_py_set_str32_to_unordered_set_u32string_multiple (
    TestResults & test_results,
    size_t repeat )
```

Invoke `test_py_set_str32_to_unordered_set_u32string_multiple()` with different size values and containers.

9.24.2.67 test_perf_py_set_str_to_unordered_set_string_multiple()

```
int test_perf_py_set_str_to_unordered_set_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

Invoke `test_py_set_str_to_unordered_set_string_multiple()` with different size values and containers.

9.24.2.68 test_perf_py_set_to_unordered_set_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_perf_py_set_to_unordered_set_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.69 test_perf_py_tuple_to_list_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_perf_py_tuple_to_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.70 test_perf_py_tuple_to_list_string_multiple()

```
int test_perf_py_tuple_to_list_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.71 test_perf_py_tuple_to_list_vector_char_multiple()

```
int test_perf_py_tuple_to_list_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.72 test_perf_py_tuple_to_vector_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_perf_py_tuple_to_vector_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.73 test_perf_py_tuple_to_vector_string_multiple()

```
int test_perf_py_tuple_to_vector_string_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.74 test_perf_py_tuple_to_vector_vector_char_multiple()

```
int test_perf_py_tuple_to_vector_vector_char_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.75 test_perf_unordered_set_string_to_py_set_multiple()

```
int test_perf_unordered_set_string_to_py_set_multiple (
    TestResults & test_results,
    size_t repeat )
```

Invoke `test_unordered_set_string_to_py_set_multiple()` with different size values and containers.

9.24.2.76 test_perf_unordered_set_to_py_set_multiple()

```
template<typename T >
int test_perf_unordered_set_to_py_set_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.77 test_perf_unordered_set_u16string_to_py_set_multiple()

```
int test_perf_unordered_set_u16string_to_py_set_multiple (
    TestResults & test_results,
    size_t repeat )
```

Invoke `test_unordered_set_u16string_to_py_set_multiple()` with different size values and containers.

9.24.2.78 test_perf_unordered_set_u32string_to_py_set_multiple()

```
int test_perf_unordered_set_u32string_to_py_set_multiple (
    TestResults & test_results,
    size_t repeat )
```

Invoke `test_unordered_set_u32string_to_py_set_multiple()` with different size values and containers.

9.24.2.79 test_perf_unordered_set_vector_char_to_py_set_multiple()

```
int test_perf_unordered_set_vector_char_to_py_set_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.80 test_perf_vector_string_to_py_list_multiple()

```
int test_perf_vector_string_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```


9.24.2.81 test_perf_vector_string_to_py_tuple_multiple()

```
int test_perf_vector_string_to_py_tuple_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.82 test_perf_vector_to_py_list_multiple()

```
template<typename T >
int test_perf_vector_to_py_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.83 test_perf_vector_to_py_tuple_multiple()

```
template<typename T >
int test_perf_vector_to_py_tuple_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t repeat )
```

9.24.2.84 test_perf_vector_u16string_to_py_list_multiple()

```
int test_perf_vector_u16string_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.85 test_perf_vector_u32string_to_py_list_multiple()

```
int test_perf_vector_u32string_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.86 test_perf_vector_vector_char_to_py_list_multiple()

```
int test_perf_vector_vector_char_to_py_list_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.87 test_perf_vector_vector_char_to_py_tuple_multiple()

```
int test_perf_vector_vector_char_to_py_tuple_multiple (
    TestResults & test_results,
    size_t repeat )
```

9.24.2.88 test_performance_all()

```
void test_performance_all (
    TestResults & test_results )
```

9.24.2.89 test_py_bool_to_cpp_bool_multiple()

```
int test_py_bool_to_cpp_bool_multiple (
    TestResults & test_results,
    size_t test_count,
    size_t repeat )
```

9.24.2.90 test_py_bytes_to_cpp_vector_char_multiple()

```
int test_py_bytes_to_cpp_vector_char_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.91 test_py_complex_to_cpp_complex_multiple()

```
int test_py_complex_to_cpp_complex_multiple (
    TestResults & test_results,
    size_t size,
    size_t repeat )
```

9.24.2.92 test_py_dict_to_cpp_std_map_like_multiple()

```
template<template< typename ... > class MapLike, typename K , typename V , PyObject *(*)(const
K &) Convert_K, PyObject *(*)(const V &) Convert_V>
int test_py_dict_to_cpp_std_map_like_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.93 test_py_dict_to_cpp_std_map_like_string_multiple()

```
template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.94 test_py_dict_to_cpp_std_map_like_vector_char_multiple()

```
template<template< typename ... > class MapLike>
int test_py_dict_to_cpp_std_map_like_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.95 test_py_dict_to_cpp_std_map_multiple()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V>
int test_py_dict_to_cpp_std_map_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.96 test_py_dict_to_cpp_std_map_string_multiple()

```
int test_py_dict_to_cpp_std_map_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.97 test_py_dict_to_cpp_std_map_vector_char_multiple()

```
int test_py_dict_to_cpp_std_map_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.98 test_py_dict_to_cpp_std_unordered_map_multiple()

```
template<typename K , typename V , PyObject *(*)(const K &) Convert_K, PyObject *(*)(const V
&) Convert_V>
int test_py_dict_to_cpp_std_unordered_map_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.99 test_py_dict_to_cpp_std_unordered_map_string_multiple()

```
int test_py_dict_to_cpp_std_unordered_map_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.100 test_py_dict_to_cpp_std_unordered_map_vector_char_multiple()

```
int test_py_dict_to_cpp_std_unordered_map_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.101 test_py_float_to_cpp_double_multiple()

```
int test_py_float_to_cpp_double_multiple (
    TestResults & test_results,
    size_t size,
    size_t repeat )
```

9.24.2.102 test_py_int_to_cpp_long_multiple()

```
int test_py_int_to_cpp_long_multiple (
    TestResults & test_results,
    size_t size,
    size_t repeat )
```

9.24.2.103 test_py_list_bytes_to_list_like_vector_char_multiple()

```
template<template< typename ... > class ListLike>
int test_py_list_bytes_to_list_like_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.104 test_py_list_bytes_to_list_vector_char_multiple()

```
int test_py_list_bytes_to_list_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.105 test_py_list_bytes_to_vector_vector_char_multiple()

```
int test_py_list_bytes_to_vector_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.106 test_py_list_str16_to_list_like_u16string_multiple()

```
template<template< typename ... > class ListLike>
int test_py_list_str16_to_list_like_u16string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.107 test_py_list_str16_to_list_u16string_multiple()

```
int test_py_list_str16_to_list_u16string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.108 test_py_list_str16_to_vector_u16string_multiple()

```
int test_py_list_str16_to_vector_u16string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.109 test_py_list_str32_to_list_like_u32string_multiple()

```
template<template< typename ... > class ListLike>
int test_py_list_str32_to_list_like_u32string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.110 test_py_list_str32_to_list_u32string_multiple()

```
int test_py_list_str32_to_list_u32string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.111 test_py_list_str32_to_vector_u32string_multiple()

```
int test_py_list_str32_to_vector_u32string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.112 test_py_list_str_to_list_like_string_multiple()

```
template<template< typename ... > class ListLike>
int test_py_list_str_to_list_like_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.113 test_py_list_str_to_list_string_multiple()

```
int test_py_list_str_to_list_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.114 test_py_list_str_to_vector_string_multiple()

```
int test_py_list_str_to_vector_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.115 test_py_list_to_list_like_multiple()

```
template<template< typename ... > class ListLike, typename T , PyObject *(*)(const T &)
ConvertCppToPy>
int test_py_list_to_list_like_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.116 test_py_list_to_list_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_list_to_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.117 test_py_list_to_vector_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_list_to_vector_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.118 test_py_set_bytes_to_unordered_set_vector_char_multiple()

```
int test_py_set_bytes_to_unordered_set_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.119 test_py_set_str16_to_unordered_set_u16string_multiple()

```
int test_py_set_str16_to_unordered_set_u16string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

Create a Python set of strings with `new_py_set_string()` Then for repeat times time the conversion of this Python object to a C++ object with `Python_Cpp_Containers::py_set_to_cpp_std_unordered_set()`

9.24.2.120 test_py_set_str32_to_unordered_set_u32string_multiple()

```
int test_py_set_str32_to_unordered_set_u32string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

Create a Python set of strings with `new_py_set_string()` Then for repeat times time the conversion of this Python object to a C++ object with `Python_Cpp_Containers::py_set_to_cpp_std_unordered_set()`

9.24.2.121 test_py_set_str_to_unordered_set_string_multiple()

```
int test_py_set_str_to_unordered_set_string_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

Create a Python set of strings with `new_py_set_string()` Then for repeat times time the conversion of this Python object to a C++ object with `Python_Cpp_Containers::py_set_to_cpp_std_unordered_set()`

9.24.2.122 test_py_set_to_unordered_set_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_set_to_unordered_set_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```


9.24.2.123 test_py_str16_to_cpp_u16string_multiple()

```
int test_py_str16_to_cpp_u16string_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.124 test_py_str32_to_cpp_u32string_multiple()

```
int test_py_str32_to_cpp_u32string_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.125 test_py_str_to_cpp_string_multiple()

```
int test_py_str_to_cpp_string_multiple (
    TestResults & test_results,
    size_t string_size,
    size_t test_count,
    size_t repeat )
```

9.24.2.126 test_py_tuple_bytes_to_list_like_vector_char_multiple()

```
template<template< typename ... > class ListLike>
int test_py_tuple_bytes_to_list_like_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.127 test_py_tuple_bytes_to_list_vector_char_multiple()

```
int test_py_tuple_bytes_to_list_vector_char_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.128 test_py_tuple_bytes_to_vector_vector_char_multiple()

```
int test_py_tuple_bytes_to_vector_vector_char_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.129 test_py_tuple_str_to_list_like_string_multiple()

```
template<template< typename ... > class ListLike>
int test_py_tuple_str_to_list_like_string_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.130 test_py_tuple_str_to_list_string_multiple()

```
int test_py_tuple_str_to_list_string_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.131 test_py_tuple_str_to_vector_string_multiple()

```
int test_py_tuple_str_to_vector_string_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.132 test_py_tuple_to_list_like_multiple()

```
template<template< typename ... > class ListLike, typename T , PyObject *(*)(const T &)
ConvertCppToPy>
int test_py_tuple_to_list_like_multiple (
    TestResultsS & test_results,
    const std::string & type,
    size_t size,
    size_t repeat,
    const std::string & container_type )
```

9.24.2.133 test_py_tuple_to_list_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_tuple_to_list_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.134 test_py_tuple_to_vector_multiple()

```
template<typename T , PyObject *(*)(const T &) ConvertCppToPy>
int test_py_tuple_to_vector_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.135 test_unordered_set_string_to_py_set_multiple()

```
int test_unordered_set_string_to_py_set_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

Create a `std::unordered_set<std::string>` with unique strings. Then for repeat times time the conversion of this C++ object to a Python object with `Python_Cpp_Containers::cpp_std_unordered_set_to_py_set()`

9.24.2.136 test_unordered_set_to_py_set_multiple()

```
template<typename T >
int test_unordered_set_to_py_set_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.137 test_unordered_set_u16string_to_py_set_multiple()

```
int test_unordered_set_u16string_to_py_set_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

Create a `std::unordered_set<std::u16string>` with unique strings. Then for repeat times time the conversion of this C++ object to a Python object with `Python_Cpp_Containers::cpp_std_unordered_set_to_py_set`

9.24.2.138 test_unordered_set_u32string_to_py_set_multiple()

```
int test_unordered_set_u32string_to_py_set_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

Create a `std::unordered_set<std::u32string>` with unique strings. Then for repeat times time the conversion of this C++ object to a Python object with `Python_Cpp_Containers::cpp_std_unordered_set_to_py_set`

9.24.2.139 test_unordered_set_vector_char_to_py_set_multiple()

```
int test_unordered_set_vector_char_to_py_set_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.140 test_vector_string_to_py_list_multiple()

```
int test_vector_string_to_py_list_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.141 test_vector_string_to_py_tuple_multiple()

```
int test_vector_string_to_py_tuple_multiple (
    TestResultsS & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.142 test_vector_to_py_list_multiple()

```
template<typename T >
int test_vector_to_py_list_multiple (
    TestResultsS & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.143 test_vector_to_py_tuple_multiple()

```
template<typename T >
int test_vector_to_py_tuple_multiple (
    TestResults & test_results,
    const std::string & type,
    size_t size,
    size_t repeat )
```

9.24.2.144 test_vector_u16string_to_py_list_multiple()

```
int test_vector_u16string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.145 test_vector_u32string_to_py_list_multiple()

```
int test_vector_u32string_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.146 test_vector_vector_char_to_py_list_multiple()

```
int test_vector_vector_char_to_py_list_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.2.147 test_vector_vector_char_to_py_tuple_multiple()

```
int test_vector_vector_char_to_py_tuple_multiple (
    TestResults & test_results,
    size_t size,
    size_t str_len,
    size_t repeat )
```

9.24.3 Variable Documentation

9.24.3.1 INC_SIZE_OF_CONTAINER_MULTIPLE

```
const size_t INC_SIZE_OF_CONTAINER_MULTIPLE = 2
```

9.24.3.2 INC_STRING_LENGTH_MULTIPLE

```
const size_t INC_STRING_LENGTH_MULTIPLE = 8
```

9.24.3.3 LIMIT_SIZE_OF_CONTAINER

```
const size_t LIMIT_SIZE_OF_CONTAINER = 1 << 21
```

9.24.3.4 LIMIT_SIZE_OF_CONTAINER_DICT

```
const size_t LIMIT_SIZE_OF_CONTAINER_DICT = 1 << 21
```

9.24.3.5 LIMIT_STRING_LENGTH

```
const size_t LIMIT_STRING_LENGTH = 1024 * 2
```

9.24.3.6 MIN_SIZE_OF_CONTAINER

```
const size_t MIN_SIZE_OF_CONTAINER = 1
```

Performance tests of one way conversions timed in C++. Created by Paul Ross on 22/05/2021.

Note: TEST_FOR_PY_ERR_ON_ENTRY and TEST_FOR_PY_ERR_ON_EXIT are not used due to the design of these tests (mainly the int return value is ignored). Those macros are used in functional and memory tests.

9.24.3.7 MIN_STRING_LENGTH_HASHABLE

```
const size_t MIN_STRING_LENGTH_HASHABLE = 16
```

9.24.3.8 MIN_STRING_LENGTH_NON_HASHABLE

```
const size_t MIN_STRING_LENGTH_NON_HASHABLE = 2
```

9.24.3.9 TEST_REPEAT

```
const size_t TEST_REPEAT = 5
```

9.25 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.h File Reference

```
#include "test_common.h"
```

Functions

- void [test_performance_all](#) ([TestResultS](#) &test_results)

9.25.1 Function Documentation

9.25.1.1 test_performance_all()

```
void test_performance_all (  
    TestResultS & test_results )
```

9.26 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cPyCppContainers.cpp File Reference

```
#include <Python.h>  
#include "structmember.h"  
#include "cpy/python_convert.h"
```

Macros

- `#define PY_SSIZE_T_CLEAN`
- `#define SINGLE_ARGUMENT_METHOD(name, doc) { #name, name, METH_O, doc }`

Functions

- static PyObject * `new_bytes` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_str` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_str16` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_str32` (PyObject *Py_UNUSED(module), PyObject *arg)
- static void `vector_double_x2` (std::vector< double > &vec)
- static PyObject * `list_x2` (PyObject *Py_UNUSED(module), PyObject *arg)
- template<typename T >
static std::vector< T > `reverse_vector` (const std::vector< T > &input)
- static PyObject * `tuple_reverse` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `dict_inc` (PyObject *Py_UNUSED(module), PyObject *arg)
- template<template< typename ... > class List, typename T >
static PyObject * `new_list` (PyObject *arg)
- static PyObject * `new_list_vector_bool` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_float` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_int` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_complex` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_bytes` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_str` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_str16` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_vector_str32` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_bool` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_float` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_int` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_complex` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_bytes` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_str` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_str16` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_list_list_str32` (PyObject *Py_UNUSED(module), PyObject *arg)
- template<typename K >
static PyObject * `new_set` (PyObject *arg)
- static PyObject * `new_set_int` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_set_float` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_set_complex` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_set_bytes` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_set_str` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_set_str16` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_set_str32` (PyObject *Py_UNUSED(module), PyObject *arg)
- template<typename K >
static PyObject * `new_frozenset` (PyObject *arg)
- static PyObject * `new_frozenset_int` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_frozenset_float` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_frozenset_complex` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_frozenset_bytes` (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * `new_frozenset_str` (PyObject *Py_UNUSED(module), PyObject *arg)
- template<template< typename ... > class Map, typename K, typename V >
static PyObject * `new_dict` (PyObject *arg)

- static PyObject * [new_dict_from_std_unordered_map_int_int](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_float_float](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_complex_complex](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_bytes_bytes](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_str_str](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_str16_str16](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_str32_str32](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_int_str](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_int_str16](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_unordered_map_int_str32](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_int_int](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_float_float](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_complex_complex](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_bytes_bytes](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_str_str](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_str16_str16](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_str32_str32](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_int_str](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_int_str16](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_from_std_map_int_str32](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- template<typename K , typename V >
static PyObject * [new_dict_debug](#) (PyObject *arg)
- static PyObject * [new_dict_debug_int_int](#) (PyObject *Py_UNUSED(module), PyObject *arg)
- static PyObject * [new_dict_debug_float_float](#) (PyObject *Py_UNUSED(module), PyObject *dict)
- PyMODINIT_FUNC [PyInit_cPyCppContainers](#) (void)

Variables

- static PyMethodDef [cPyCppContainersMethods](#) []
- static struct PyModuleDef [cPyCppContainersmodule](#)

9.26.1 Macro Definition Documentation

9.26.1.1 PY_SSIZE_T_CLEAN

```
#define PY_SSIZE_T_CLEAN
```

Created by Paul Ross on 18/06/2021.

9.26.1.2 SINGLE_ARGUMENT_METHOD

```
#define SINGLE_ARGUMENT_METHOD (
    name,
    doc ) { #name, name, METH_O, doc }
```

9.26.2 Function Documentation

9.26.2.1 dict_inc()

```
static PyObject* dict_inc (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Creates a new dict[bytes, int] with the values incremented by 1 in C++.

Parameters

<i>_unused_module</i>	
<i>arg</i>	The Python dictionary of [bytes, int] to increment in C++. This is const.

Returns

A new Python dict of [bytes, int] with the values incremented.

9.26.2.2 list_x2()

```
static PyObject* list_x2 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new list of floats with doubled values.

Parameters

<i>_unused_module</i>	
<i>arg</i>	The Python list of floats. This is const.

Returns

A new Python list of floats with the values doubled.

9.26.2.3 new_bytes()

```
static PyObject* new_bytes (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Take a Python bytes object, convert it to a `std::vector<char>` then convert that back to a Python bytes object.

9.26.2.4 new_dict()

```
template<template< typename ... > class Map, typename K , typename V >
static PyObject* new_dict (
    PyObject * arg ) [static]
```

Create a new dict of [K, V] by copying into a `std::unordered_map` and back.

Parameters

<i>arg</i>	The Python dict. This is const.
------------	---------------------------------

Returns

A new Python dict of [K, V].

9.26.2.5 new_dict_debug()

```
template<typename K , typename V >
static PyObject* new_dict_debug (
    PyObject * arg ) [static]
```

Create a new dict of [K, V] by copying into a `std::unordered_map`.

Parameters

<i>arg</i>	The Python dict. This is const.
------------	---------------------------------

Returns

None.

9.26.2.6 new_dict_debug_float_float()

```
static PyObject* new_dict_debug_float_float (
    PyObject * Py_UNUSED(module),
    PyObject * dict ) [static]
```

9.26.2.7 new_dict_debug_int_int()

```
static PyObject* new_dict_debug_int_int (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.8 new_dict_from_std_map_bytes_bytes()

```
static PyObject* new_dict_from_std_map_bytes_bytes (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.9 new_dict_from_std_map_complex_complex()

```
static PyObject* new_dict_from_std_map_complex_complex (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.10 new_dict_from_std_map_float_float()

```
static PyObject* new_dict_from_std_map_float_float (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.11 new_dict_from_std_map_int_int()

```
static PyObject* new_dict_from_std_map_int_int (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.12 new_dict_from_std_map_int_str()

```
static PyObject* new_dict_from_std_map_int_str (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.13 new_dict_from_std_map_int_str16()

```
static PyObject* new_dict_from_std_map_int_str16 (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.14 new_dict_from_std_map_int_str32()

```
static PyObject* new_dict_from_std_map_int_str32 (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.15 new_dict_from_std_map_str16_str16()

```
static PyObject* new_dict_from_std_map_str16_str16 (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.16 new_dict_from_std_map_str32_str32()

```
static PyObject* new_dict_from_std_map_str32_str32 (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.17 new_dict_from_std_map_str_str()

```
static PyObject* new_dict_from_std_map_str_str (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.18 new_dict_from_std_unordered_map_bytes_bytes()

```
static PyObject* new_dict_from_std_unordered_map_bytes_bytes (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.19 new_dict_from_std_unordered_map_complex_complex()

```
static PyObject* new_dict_from_std_unordered_map_complex_complex (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.20 new_dict_from_std_unordered_map_float_float()

```
static PyObject* new_dict_from_std_unordered_map_float_float (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.21 new_dict_from_std_unordered_map_int_int()

```
static PyObject* new_dict_from_std_unordered_map_int_int (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.22 new_dict_from_std_unordered_map_int_str()

```
static PyObject* new_dict_from_std_unordered_map_int_str (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.23 new_dict_from_std_unordered_map_int_str16()

```
static PyObject* new_dict_from_std_unordered_map_int_str16 (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.24 new_dict_from_std_unordered_map_int_str32()

```
static PyObject* new_dict_from_std_unordered_map_int_str32 (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.26.2.25 new_dict_from_std_unordered_map_str16_str16()

```
static PyObject* new_dict_from_std_unordered_map_str16_str16 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.26 new_dict_from_std_unordered_map_str32_str32()

```
static PyObject* new_dict_from_std_unordered_map_str32_str32 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.27 new_dict_from_std_unordered_map_str_str()

```
static PyObject* new_dict_from_std_unordered_map_str_str (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.28 new_frozenset()

```
template<typename K >
static PyObject* new_frozenset (
    PyObject * arg ) [static]
```

Create a new frozenset of [K] by copying into a `std::unordered_set` and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python frozenset of [K].

9.26.2.29 new_frozenset_bytes()

```
static PyObject* new_frozenset_bytes (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new frozenset of [bytes] by copying into a `std::unordered_set` and back.

Parameters

<i>arg</i>	The Python frozenset. This is const.
------------	--------------------------------------

Returns

A new Python frozenset of [bytes].

9.26.2.30 new_frozenset_complex()

```
static PyObject* new_frozenset_complex (  
    PyObject * Py_UNUSED(module),  
    PyObject * arg ) [static]
```

Create a new frozenset of [complex] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python frozenset. This is const.
------------	--------------------------------------

Returns

A new Python frozenset of [complex].

9.26.2.31 new_frozenset_float()

```
static PyObject* new_frozenset_float (  
    PyObject * Py_UNUSED(module),  
    PyObject * arg ) [static]
```

Create a new frozenset of [float] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python frozenset. This is const.
------------	--------------------------------------

Returns

A new Python frozenset of [float].

9.26.2.32 new_frozenset_int()

```
static PyObject* new_frozenset_int (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new frozenset of [int] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python frozenset. This is const.
------------	--------------------------------------

Returns

A new Python frozenset of [int].

9.26.2.33 new_frozenset_str()

```
static PyObject* new_frozenset_str (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new frozenset of [str] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python frozenset. This is const.
------------	--------------------------------------

Returns

A new Python frozenset of [str].

9.26.2.34 new_list()

```
template<template< typename ... > class List, typename T >
static PyObject* new_list (
    PyObject * arg ) [static]
```

Create a new list of T by copying into a vector and back.

Parameters

<i>arg</i>	The Python list. This is const.
------------	---------------------------------

Returns

A new Python list of T.

9.26.2.35 new_list_list_bool()

```
static PyObject* new_list_list_bool (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.36 new_list_list_bytes()

```
static PyObject* new_list_list_bytes (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.37 new_list_list_complex()

```
static PyObject* new_list_list_complex (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.38 new_list_list_float()

```
static PyObject* new_list_list_float (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.39 new_list_list_int()

```
static PyObject* new_list_list_int (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.40 new_list_list_str()

```
static PyObject* new_list_list_str (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.41 new_list_list_str16()

```
static PyObject* new_list_list_str16 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.42 new_list_list_str32()

```
static PyObject* new_list_list_str32 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.43 new_list_vector_bool()

```
static PyObject* new_list_vector_bool (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new list of bools by copying into a vector and back.

Parameters

<i>arg</i>	The Python list. This is const.
------------	---------------------------------

Returns

A new Python list of bool.

9.26.2.44 new_list_vector_bytes()

```
static PyObject* new_list_vector_bytes (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.45 new_list_vector_complex()

```
static PyObject* new_list_vector_complex (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.46 new_list_vector_float()

```
static PyObject* new_list_vector_float (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.47 new_list_vector_int()

```
static PyObject* new_list_vector_int (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.48 new_list_vector_str()

```
static PyObject* new_list_vector_str (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.49 new_list_vector_str16()

```
static PyObject* new_list_vector_str16 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.50 new_list_vector_str32()

```
static PyObject* new_list_vector_str32 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

9.26.2.51 new_set()

```
template<typename K >
static PyObject* new_set (
    PyObject * arg ) [static]
```

Create a new set of [K] by copying into a `std::unordered_set` and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of [K].

9.26.2.52 new_set_bytes()

```
static PyObject* new_set_bytes (  
    PyObject * Py_UNUSED(module),  
    PyObject * arg ) [static]
```

Create a new set of [bytes] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of [bytes].

9.26.2.53 new_set_complex()

```
static PyObject* new_set_complex (  
    PyObject * Py_UNUSED(module),  
    PyObject * arg ) [static]
```

Create a new set of [complex] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of [complex].

9.26.2.54 new_set_float()

```
static PyObject* new_set_float (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new set of [float] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of [float].

9.26.2.55 new_set_int()

```
static PyObject* new_set_int (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new set of [int] by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of [int].

9.26.2.56 new_set_str()

```
static PyObject* new_set_str (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new set of str by copying into a std::unordered_set and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of [str].

9.26.2.57 new_set_str16()

```
static PyObject* new_set_str16 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new set of str with 16 bit characters by copying into a `std::unordered_set` and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of str16.

9.26.2.58 new_set_str32()

```
static PyObject* new_set_str32 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Create a new set of str with 32 bit characters by copying into a `std::unordered_set` and back.

Parameters

<i>arg</i>	The Python set. This is const.
------------	--------------------------------

Returns

A new Python set of str32.

9.26.2.59 new_str()

```
static PyObject* new_str (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Take a Python `str` object (8 bit characters), convert it to a `std::string` then convert that back to a Python `str` object.

9.26.2.60 new_str16()

```
static PyObject* new_str16 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Take a Python `str` object (16 bit characters), convert it to a `std::string` then convert that back to a Python `str` object.

9.26.2.61 new_str32()

```
static PyObject* new_str32 (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Take a Python `str` object (32 bit characters), convert it to a `std::string` then convert that back to a Python `str` object.

9.26.2.62 PyInit_cPyCppContainers()

```
PyMODINIT_FUNC PyInit_cPyCppContainers (
    void )
```

The Python module initialisation.

9.26.2.63 reverse_vector()

```
template<typename T >
static std::vector<T> reverse_vector (
    const std::vector< T > & input ) [static]
```

Returns a new vector reversed.

Template Parameters

<i>T</i>	The type of the members of the vector.
----------	--

Parameters

<i>input</i>	The vector to be reversed.
--------------	----------------------------

Returns

A new vector reversed.

9.26.2.64 tuple_reverse()

```
static PyObject* tuple_reverse (
    PyObject * Py_UNUSED(module),
    PyObject * arg ) [static]
```

Reverse a tuple of bytes in C++.

Parameters

<code>_unused_module</code>	
<code>arg</code>	The Python tuple of bytes to be reversed.

Returns

A new tuple of bytes reversed.

9.26.2.65 vector_double_x2()

```
static void vector_double_x2 (
    std::vector< double > & vec ) [static]
```

Double the values of a vector in-place.

Parameters

<code>vec</code>	The vector to double.
------------------	-----------------------

9.26.3 Variable Documentation

9.26.3.1 cPyCppContainersMethods

```
PyMethodDef cPyCppContainersMethods[] [static]
```

The Python Extension methods.

9.26.3.2 cPyCppContainersmodule

```
struct PyModuleDef cPyCppContainersmodule [static]
```

Initial value:

```
= {
    PyModuleDef_HEAD_INIT,
```

```

    "cPyCppContainers",
    "Example extension module that converts Python containers to and from their C++ equivalents.",
    -1,
    cPyCppContainersMethods,
    NULL,
    NULL,
    NULL,
    NULL
}

```

The Python extension definition.

9.27 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.cpp File Reference

```

#include <Python.h>
#include "structmember.h"
#include "cUserDefined.h"
#include "cpy/python_object_convert.h"

```

Classes

- struct [CustomObject](#)

Namespaces

- [Python_Cpp_Containers](#)
Conversion functions for individual Python objects.

Macros

- #define [PY_SSIZE_T_CLEAN](#)

Functions

- static void [Custom_dealloc](#) ([CustomObject](#) *self)
- static PyObject * [Custom_new](#) (PyTypeObject *type, PyObject *Py_UNUSED(args), PyObject *Py_UNUSED(kwds))
- static int [Custom_init](#) ([CustomObject](#) *self, PyObject *args, PyObject *kwds)
- static PyObject * [Custom_getfirst](#) ([CustomObject](#) *self, void *Py_UNUSED(closure))
- static int [Custom_setfirst](#) ([CustomObject](#) *self, PyObject *value, void *Py_UNUSED(closure))
- static PyObject * [Custom_getlast](#) ([CustomObject](#) *self, void *Py_UNUSED(closure))
- static int [Custom_setlast](#) ([CustomObject](#) *self, PyObject *value, void *Py_UNUSED(closure))
- static PyObject * [Custom_name](#) ([CustomObject](#) *self, PyObject *Py_UNUSED(ignored))
- int [py_custom_object_check](#) (PyObject *op)
- [CppCustomObject](#) [py_custom_object_to_cpp_custom_object](#) (PyObject *op)
- PyObject * [cpp_custom_object_to_py_custom_object](#) (const [CppCustomObject](#) &obj)
- template<T> PyObject * [Python_Cpp_Containers::cpp_std_list_like_to_py_list](#)< [CppCustomObject](#) > (const std::vector< [CppCustomObject](#) > &container)

- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< CppCustomObject > (PyObject *op, std::vector< CppCustomObject > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject > (const std::map< long, CppCustomObject > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject > (PyObject *op, std::map< long, CppCustomObject > &map)`
- `static PyObject * reverse_list_names (PyObject *Py_UNUSED(module), PyObject *arg)`
- `static PyObject * reverse_dict_names (PyObject *Py_UNUSED(module), PyObject *arg)`
- `PyMODINIT_FUNC Pylnit_cUserDefined (void)`

Variables

- `static PyMemberDef Custom_members []`
- `static PyGetSetDef Custom_getsetters []`
- `static PyMethodDef Custom_methods []`
- `static PyTypeObject CustomType`
- `static PyMethodDef cUserDefinedMethods []`
- `static struct PyModuleDef cUserDefinedmodule`

9.27.1 Macro Definition Documentation

9.27.1.1 PY_SSIZE_T_CLEAN

```
#define PY_SSIZE_T_CLEAN
```

9.27.2 Function Documentation

9.27.2.1 cpp_custom_object_to_py_custom_object()

```
PyObject* cpp_custom_object_to_py_custom_object (
    const CppCustomObject & obj )
```

9.27.2.2 Custom_dealloc()

```
static void Custom_dealloc (
    CustomObject * self ) [static]
```

9.27.2.3 Custom_getfirst()

```
static PyObject* Custom_getfirst (
    CustomObject * self,
    void * Py_UNUSEDclosure ) [static]
```

9.27.2.4 Custom_getlast()

```
static PyObject* Custom_getlast (
    CustomObject * self,
    void * Py_UNUSEDclosure ) [static]
```

9.27.2.5 Custom_init()

```
static int Custom_init (
    CustomObject * self,
    PyObject * args,
    PyObject * kwds ) [static]
```

9.27.2.6 Custom_name()

```
static PyObject* Custom_name (
    CustomObject * self,
    PyObject * Py_UNUSEDignored ) [static]
```

9.27.2.7 Custom_new()

```
static PyObject* Custom_new (
    PyTypeObject * type,
    PyObject * Py_UNUSEDargs,
    PyObject * Py_UNUSEDkwds ) [static]
```

9.27.2.8 Custom_setfirst()

```
static int Custom_setfirst (
    CustomObject * self,
    PyObject * value,
    void * Py_UNUSEDclosure ) [static]
```

9.27.2.9 Custom_setlast()

```
static int Custom_setlast (
    CustomObject * self,
    PyObject * value,
    void * Py_UNUSEDclosure ) [static]
```

9.27.2.10 py_custom_object_check()

```
int py_custom_object_check (
    PyObject * op )
```

9.27.2.11 py_custom_object_to_cpp_custom_object()

```
CppCustomObject py_custom_object_to_cpp_custom_object (
    PyObject * op )
```

9.27.2.12 PyInit_cUserDefined()

```
PyMODINIT_FUNC PyInit_cUserDefined (
    void )
```

9.27.2.13 reverse_dict_names()

```
static PyObject* reverse_dict_names (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.27.2.14 reverse_list_names()

```
static PyObject* reverse_list_names (
    PyObject * Py_UNUSEDmodule,
    PyObject * arg ) [static]
```

9.27.3 Variable Documentation

9.27.3.1 cUserDefinedMethods

```
PyMethodDef cUserDefinedMethods[] [static]
```

Initial value:

```
= {
    {"reverse_list_names", reverse_list_names, METH_O,
     "Take a list of cUserDefined.Custom objects"
     " and return a new list with the name reversed."},
    {"reverse_dict_names", reverse_dict_names, METH_O,
     "Take a dict of [int, cUserDefined.Custom] objects"
     " and return a new dict with the name reversed."},
    {NULL, NULL, 0, NULL}
}
```

9.27.3.2 cUserDefinedmodule

```
struct PyModuleDef cUserDefinedmodule [static]
```

Initial value:

```
= {
    PyModuleDef_HEAD_INIT,
    "cUserDefined",
    "Example extension module that defines a custom object and converts to and from their C++ equivalents.",
    -1,
    cUserDefinedMethods,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL
}
```

9.27.3.3 Custom_getsetters

```
PyGetSetDef Custom_getsetters[] [static]
```

Initial value:

```
= {
    {"first", (getter) Custom_getfirst, (setter) Custom_setfirst,
     "first name", NULL},
    {"last", (getter) Custom_getlast, (setter) Custom_setlast,
     "last name", NULL},
    {NULL, NULL, NULL, NULL, NULL}
}
```

9.27.3.4 Custom_members

```
PyMemberDef Custom_members[] [static]
```

Initial value:

```
= {
    {"number", T_INT, offsetof(CustomObject, number), 0,
     "custom number"},
    {NULL, 0, 0, 0, NULL}
}
```

9.27.3.5 Custom_methods

```
PyMethodDef Custom_methods[] [static]
```

Initial value:

```
= {  
    {"name", (PyCFunction) Custom_name, METH_NOARGS,  
     "Return the name, combining the first and last name"  
    },  
    {NULL, NULL, 0, NULL}  
}
```

9.27.3.6 CustomType

```
PyTypeObject CustomType [static]
```

Initial value:

```
= {  
    PyVarObject_HEAD_INIT(NULL, 0)  
    .tp_name = "cUserDefined.Custom",  
    .tp_basicsize = sizeof(CustomObject),  
    .tp_itemsize = 0,  
    .tp_dealloc = (destructor) Custom_dealloc,  
    .tp_flags = Py_TPFLAGS_DEFAULT | Py_TPFLAGS_BASETYPE,  
    .tp_doc = "Custom objects",  
    .tp_methods = Custom_methods,  
    .tp_members = Custom_members,  
    .tp_getset = Custom_getsetters,  
    .tp_init = (initproc) Custom_init,  
    .tp_new = Custom_new,  
}
```

9.28 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/cUserDefined.h File Reference

```
#include <string>  
#include <utility>  
#include "cpy/python_convert.h"
```

Classes

- class [CppCustomObject](#)

Namespaces

- [Python_Cpp_Containers](#)

Conversion functions for individual Python objects.

Functions

- `template<> PyObject * Python_Cpp_Containers::cpp_std_list_like_to_py_list< CppCustomObject > (const std::vector< CppCustomObject > &container)`
- `template<> int Python_Cpp_Containers::py_list_to_cpp_std_list_like< CppCustomObject > (PyObject *op, std::vector< CppCustomObject > &container)`
- `template<> PyObject * Python_Cpp_Containers::cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject > (const std::map< long, CppCustomObject > &map)`
- `template<> int Python_Cpp_Containers::py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject > (PyObject *op, std::map< long, CppCustomObject > &map)`

9.29 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ext/custom_3_Python3.9.0.c File Reference

9.30 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/main.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <Python.h>
#include <cpp/save_stream_state.h>
#include "cpy/tests/test_functional.h"
#include "cpy/tests/test_internal.h"
#include "cpy/tests/test_performance.h"
#include "cpy/tests/test_memory.h"
```

Macros

- `#define TEST_INTERNAL_ALL 1`
- `#define TEST_FUNCTIONAL_ALL 1`
- `#define TEST_PERFORMANCE_ALL 1`
- `#define TEST_MEMORY_ALL 1`

Functions

- `int test_all ()`
- `void explore_hash_reserve ()`
- `int main ()`

9.30.1 Macro Definition Documentation

9.30.1.1 TEST_FUNCTIONAL_ALL

```
#define TEST_FUNCTIONAL_ALL 1
```

9.30.1.2 TEST_INTERNAL_ALL

```
#define TEST_INTERNAL_ALL 1
```

9.30.1.3 TEST_MEMORY_ALL

```
#define TEST_MEMORY_ALL 1
```

9.30.1.4 TEST_PERFORMANCE_ALL

```
#define TEST_PERFORMANCE_ALL 1
```

9.30.2 Function Documentation

9.30.2.1 explore_hash_reserve()

```
void explore_hash_reserve ( )
```

9.30.2.2 main()

```
int main ( )
```

9.30.2.3 test_all()

```
int test_all ( )
```

9.31 /Users/paulross/CLionProjects/PythonCppHomogeneous↔ Containers/src/py/code_gen.py File Reference

Classes

- class [src.py.code_gen.CodeCount](#)

Namespaces

- [src.py.code_gen](#)

Functions

- str [src.py.code_gen.defn_name_from_decl_name](#) (str name, str cpp_container)
- CodeCount [src.py.code_gen.unary_declarations](#) ()
- CodeCount [src.py.code_gen.unary_definitions](#) ()
- CodeCount [src.py.code_gen.dict_map_declarations](#) ()
- CodeCount [src.py.code_gen.dict_map_definitions](#) ()
- CodeCount [src.py.code_gen.declarations](#) ()
- CodeCount [src.py.code_gen.definitions](#) ()
- None [src.py.code_gen.write_files](#) ()
- def [src.py.code_gen.main](#) ()

Variables

- [src.py.code_gen.logger](#) = logging.getLogger(__file__)
- string [src.py.code_gen.CPP_NAMESPACE](#) = 'Python_Cpp_Containers'
- string [src.py.code_gen.PROJECT_VERSION](#) = '0.4.0'
- dictionary [src.py.code_gen.CPP_TYPE_TO_FUNCS](#)
- tuple [src.py.code_gen.UNARY_COLLECTIONS](#)
- list [src.py.code_gen.REQUIRED_INCLUDES](#)
- dictionary [src.py.code_gen.CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINER](#)
- string [src.py.code_gen.CPP_UNARY_FUNCTION_TO_PY_BASE_DECL](#)
- string [src.py.code_gen.CPP_UNARY_FUNCTION_TO_PY_DECL](#)
- string [src.py.code_gen.CPP_UNARY_FUNCTION_TO_PY_DEFN](#)
- string [src.py.code_gen.PY_TO_CPP_UNARY_FUNCTION_BASE_DECL](#)
- string [src.py.code_gen.PY_TO_CPP_UNARY_FUNCTION_DECL](#)
- string [src.py.code_gen.PY_TO_CPP_UNARY_FUNCTION_DEFN](#)
- tuple [src.py.code_gen.CPP_MAP_TYPES](#) = ('std::unordered_map', 'std::map')
- string [src.py.code_gen.CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL](#)
- string [src.py.code_gen.CPP_MAP_TYPE_TO_PY_DICT_DECL](#)
- string [src.py.code_gen.CPP_MAP_TYPE_TO_PY_DICT_DEFN](#)
- string [src.py.code_gen.CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL](#)
- string [src.py.code_gen.CPP_PY_DICT_TO_MAP_TYPE_DECL](#)
- string [src.py.code_gen.CPP_PY_DICT_TO_MAP_TYPE_DEFN](#)
- string [src.py.code_gen.AUTO_FILE_NAME](#) = 'auto_py_convert_internal'

9.32 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_common.py File Reference

Classes

- class [src.py.code_gen_common.CppTypeFunctions](#)
- class [src.py.code_gen_common.TypeConversionFunctions](#)
- class [src.py.code_gen_common.UnaryFunctions](#)

Namespaces

- [src.py.code_gen_common](#)

9.33 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py/code_gen_documentation.py File Reference

Namespaces

- [src.py.code_gen_documentation](#)

Functions

- [typing.List\[str\]](#) [src.py.code_gen_documentation.doxygen_cpp_to_python_unary_base_class](#) (str cpp_container, str python_container)
- [def](#) [src.py.code_gen_documentation.doxygen_cpp_to_python_unary_instantiation](#) (str cpp_container, str python_container, str cpp_type, str py_type)
- [typing.List\[str\]](#) [src.py.code_gen_documentation.doxygen_python_to_cpp_unary_base_class](#) (str cpp_container, str python_container)
- [def](#) [src.py.code_gen_documentation.doxygen_python_to_cpp_unary_instantiation](#) (str cpp_container, str python_container, str cpp_type, str py_type)
- [typing.List\[str\]](#) [src.py.code_gen_documentation.doxygen_cpp_to_python_dict_base_class](#) ()
- [def](#) [src.py.code_gen_documentation.doxygen_cpp_to_python_dict_instantiation](#) (str cpp_key_type, str cpp_val_type, str py_key_type, str py_val_type)
- [def](#) [src.py.code_gen_documentation.doxygen_python_dict_to_cpp_base_class](#) ()
- [def](#) [src.py.code_gen_documentation.doxygen_python_dict_to_cpp_instantiation](#) (str cpp_key_type, str cpp_val_type, str py_key_type, str py_val_type)
- [str](#) [src.py.code_gen_documentation.comment_str](#) (str s)
- [typing.List\[str\]](#) [src.py.code_gen_documentation.comment_list_str](#) (typing.List[str] inputs)
- [def](#) [src.py.code_gen_documentation.cpp_comment_section](#) (typing.List[str] str_list, str title, str sep)
- [typing.List\[str\]](#) [src.py.code_gen_documentation.documentation](#) (typing.Tuple[code_gen_common.UnaryFunctions,...] unary_collections, typing.Dict[str, code_gen_common.CppTypeFunctions] cpp_type_to_funcs)
- [typing.List\[str\]](#) [src.py.code_gen_documentation.get_codegen_please_no_edit_warning](#) (bool is_end)
- [def](#) [src.py.code_gen_documentation.get_codegen_please_no_edit_warning_context](#) (typing.List[str] str_list)

Variables

- [int](#) [src.py.code_gen_documentation.WIDTH](#) = 75 - len("//")

9.34 `/Users/paulross/CLionProjects/PythonCppHomogeneous↵`
`Containers/src/ReadMe.md` File
Reference

9.35 `/Users/paulross/CLionProjects/PythonCppHomogeneous↵`
`Containers/README.md` File
Reference

Index

/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/README.md,
418 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/README.md,
418 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex__init__.py,
231 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp,
236 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/TestFramework.h,
241 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/get_rss.cpp,
231 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/m
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/get_rss.h,
234 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/save_stream_state.h,
235 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/cpy/auto_py_convert_internal.cpp,
244 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/cpy/auto_py_convert_internal.h,
256 /Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/py
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/cpy/python_container_convert.cpp,
268 ~StreamFormatState
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/python_state_container_convert.h,
268 StreamFormatState, 268
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/python_convert.h,
269 atomic::TestMeanExpectTime
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/ex.cpp/python_convert_scrap.h,
271 src.py.code_gen, 198
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_convert.cpp,
273 comment_list_str
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/python_object_convert.h,
274 src.py.code_gen, documentation, 204
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.cpp,
275 compare_dict
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_common.h,
301 test_common.h, 307, 308
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.cpp,
339 test_common.h, 308
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_functional.h,
344 compare_dict< std::map, std::u16string, std::u16string
>
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.cpp,
344 test_common.h, 308
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_internal.h,
347 compare_dict< std::map, std::u32string, std::u32string
>
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.cpp,
348 test_common.h, 309
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_memory.h,
351 compare_dict< std::unordered_map, std::string,
std::string >
/Users/paulross/CLionProjects/PythonCppHomogeneousContainers/src/cpy/tests/test_performance.cpp,
test_common.h, 309

compare_dict< std::unordered_map, std::u16string,
 std::u16string >
 test_common.cpp, 278
 test_common.h, 309
 compare_dict< std::unordered_map, std::u32string,
 std::u32string >
 test_common.cpp, 278
 test_common.h, 309
 compare_list
 test_common.h, 309, 310
 compare_list< bool >
 test_common.cpp, 278
 test_common.h, 310
 compare_list< double >
 test_common.cpp, 278
 test_common.h, 311
 compare_list< long >
 test_common.cpp, 279
 test_common.h, 311
 compare_list< std::complex< double > >
 test_common.cpp, 279
 test_common.h, 311
 compare_list< std::string >
 test_common.cpp, 279
 test_common.h, 311
 compare_list< std::u16string >
 test_common.cpp, 279
 compare_list< std::u32string >
 test_common.cpp, 279
 compare_list< std::vector< char > >
 test_common.cpp, 279
 test_common.h, 311
 compare_set
 test_common.h, 311, 312
 compare_set< std::string >
 test_common.cpp, 280
 test_common.h, 313
 compare_set< std::u16string >
 test_common.cpp, 280
 test_common.h, 313
 compare_set< std::u32string >
 test_common.cpp, 280
 test_common.h, 313
 compare_set< std::vector< char > >
 test_common.cpp, 280
 test_common.h, 313
 compare_tuple
 test_common.h, 313, 314
 compare_tuple< bool >
 test_common.cpp, 280
 test_common.h, 314
 compare_tuple< double >
 test_common.cpp, 280
 test_common.h, 315
 compare_tuple< long >
 test_common.cpp, 281
 test_common.h, 315
 compare_tuple< std::complex< double > >
 test_common.cpp, 281
 test_common.h, 315
 compare_tuple< std::string >
 test_common.cpp, 281
 test_common.h, 315
 compare_tuple< std::u16string >
 test_common.cpp, 281
 test_common.h, 315
 compare_tuple< std::u32string >
 test_common.cpp, 281
 test_common.h, 315
 compare_tuple< std::vector< char > >
 test_common.cpp, 281
 test_common.h, 316
 compare_tuple_or_list
 test_common.h, 316
 count_of_unique_string
 TestFramework.cpp, 237
 TestFramework.h, 242
 cpp_bool_to_py_bool
 Python_Cpp_Containers, 29
 cpp_comment_section
 src.py.code_gen_documentation, 204
 cpp_complex_to_py_complex
 Python_Cpp_Containers, 30
 cpp_custom_object_to_py_custom_object
 cUserDefined.cpp, 409
 cpp_double_to_py_float
 Python_Cpp_Containers, 30
 cpp_long_to_py_long
 Python_Cpp_Containers, 30
 CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL
 src.py.code_gen, 198
 CPP_MAP_TYPE_TO_PY_DICT_DECL
 src.py.code_gen, 198
 CPP_MAP_TYPE_TO_PY_DICT_DEFN
 src.py.code_gen, 199
 CPP_MAP_TYPES
 src.py.code_gen, 199
 CPP_NAMESPACE
 src.py.code_gen, 199
 CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL
 src.py.code_gen, 199
 CPP_PY_DICT_TO_MAP_TYPE_DECL
 src.py.code_gen, 199
 CPP_PY_DICT_TO_MAP_TYPE_DEFN
 src.py.code_gen, 200
 cpp_std_list_like_to_py_list
 Python_Cpp_Containers, 31
 cpp_std_list_like_to_py_list< bool >
 Python_Cpp_Containers, 31, 32
 cpp_std_list_like_to_py_list< CppCustomObject >
 Python_Cpp_Containers, 32
 cpp_std_list_like_to_py_list< double >
 Python_Cpp_Containers, 32, 33
 cpp_std_list_like_to_py_list< long >
 Python_Cpp_Containers, 33

- cpp_std_list_like_to_py_list< std::complex< double >
>
Python_Cpp_Containers, [34](#)
- cpp_std_list_like_to_py_list< std::string >
Python_Cpp_Containers, [35](#)
- cpp_std_list_like_to_py_list< std::u16string >
Python_Cpp_Containers, [35](#), [36](#)
- cpp_std_list_like_to_py_list< std::u32string >
Python_Cpp_Containers, [36](#)
- cpp_std_list_like_to_py_list< std::vector< char > >
Python_Cpp_Containers, [37](#)
- cpp_std_list_like_to_py_tuple
Python_Cpp_Containers, [38](#)
- cpp_std_list_like_to_py_tuple< bool >
Python_Cpp_Containers, [38](#), [39](#)
- cpp_std_list_like_to_py_tuple< double >
Python_Cpp_Containers, [39](#), [40](#)
- cpp_std_list_like_to_py_tuple< long >
Python_Cpp_Containers, [40](#)
- cpp_std_list_like_to_py_tuple< std::complex< double
> >
Python_Cpp_Containers, [41](#)
- cpp_std_list_like_to_py_tuple< std::string >
Python_Cpp_Containers, [41](#), [42](#)
- cpp_std_list_like_to_py_tuple< std::u16string >
Python_Cpp_Containers, [42](#), [43](#)
- cpp_std_list_like_to_py_tuple< std::u32string >
Python_Cpp_Containers, [43](#)
- cpp_std_list_like_to_py_tuple< std::vector< char > >
Python_Cpp_Containers, [44](#)
- cpp_std_map_like_to_py_dict
Python_Cpp_Containers, [44](#), [45](#)
- cpp_std_map_like_to_py_dict< std::map, bool, bool >
Python_Cpp_Containers, [45](#)
- cpp_std_map_like_to_py_dict< std::map, bool, double
>
Python_Cpp_Containers, [46](#)
- cpp_std_map_like_to_py_dict< std::map, bool, long >
Python_Cpp_Containers, [46](#)
- cpp_std_map_like_to_py_dict< std::map, bool,
std::complex< double > >
Python_Cpp_Containers, [46](#)
- cpp_std_map_like_to_py_dict< std::map, bool,
std::string >
Python_Cpp_Containers, [47](#)
- cpp_std_map_like_to_py_dict< std::map, bool,
std::u16string >
Python_Cpp_Containers, [47](#)
- cpp_std_map_like_to_py_dict< std::map, bool,
std::u32string >
Python_Cpp_Containers, [48](#)
- cpp_std_map_like_to_py_dict< std::map, bool,
std::vector< char > >
Python_Cpp_Containers, [48](#)
- cpp_std_map_like_to_py_dict< std::map, double, bool
>
Python_Cpp_Containers, [48](#)
- cpp_std_map_like_to_py_dict< std::map, double, dou-
ble >
Python_Cpp_Containers, [49](#)
- cpp_std_map_like_to_py_dict< std::map, double, long
>
Python_Cpp_Containers, [49](#)
- cpp_std_map_like_to_py_dict< std::map, double,
std::complex< double > >
Python_Cpp_Containers, [50](#)
- cpp_std_map_like_to_py_dict< std::map, double,
std::string >
Python_Cpp_Containers, [50](#)
- cpp_std_map_like_to_py_dict< std::map, double,
std::u16string >
Python_Cpp_Containers, [50](#)
- cpp_std_map_like_to_py_dict< std::map, double,
std::u32string >
Python_Cpp_Containers, [51](#)
- cpp_std_map_like_to_py_dict< std::map, double,
std::vector< char > >
Python_Cpp_Containers, [51](#)
- cpp_std_map_like_to_py_dict< std::map, long, bool >
Python_Cpp_Containers, [52](#)
- cpp_std_map_like_to_py_dict< std::map, long, Cpp-
CustomObject >
Python_Cpp_Containers, [52](#)
- cpp_std_map_like_to_py_dict< std::map, long, double
>
Python_Cpp_Containers, [52](#)
- cpp_std_map_like_to_py_dict< std::map, long, long >
Python_Cpp_Containers, [52](#)
- cpp_std_map_like_to_py_dict< std::map, long,
std::complex< double > >
Python_Cpp_Containers, [53](#)
- cpp_std_map_like_to_py_dict< std::map, long,
std::string >
Python_Cpp_Containers, [53](#)
- cpp_std_map_like_to_py_dict< std::map, long,
std::u16string >
Python_Cpp_Containers, [54](#)
- cpp_std_map_like_to_py_dict< std::map, long,
std::u32string >
Python_Cpp_Containers, [54](#)
- cpp_std_map_like_to_py_dict< std::map, long,
std::vector< char > >
Python_Cpp_Containers, [54](#)
- cpp_std_map_like_to_py_dict< std::map, std::complex<
double >, bool >
Python_Cpp_Containers, [55](#)
- cpp_std_map_like_to_py_dict< std::map, std::complex<
double >, double >
Python_Cpp_Containers, [55](#)
- cpp_std_map_like_to_py_dict< std::map, std::complex<
double >, long >
Python_Cpp_Containers, [56](#)
- cpp_std_map_like_to_py_dict< std::map, std::string,
bool >
Python_Cpp_Containers, [56](#)

[illegible]

- std::vector< char >, bool >
 - Python_Cpp_Containers, [87](#)
- cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, double >
 - Python_Cpp_Containers, [88](#)
- cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, long >
 - Python_Cpp_Containers, [88](#)
- cpp_std_unordered_set_to_py_frozenset
 - Python_Cpp_Containers, [88](#)
- cpp_std_unordered_set_to_py_frozenset< bool >
 - Python_Cpp_Containers, [89](#)
- cpp_std_unordered_set_to_py_frozenset< double >
 - Python_Cpp_Containers, [89](#)
- cpp_std_unordered_set_to_py_frozenset< long >
 - Python_Cpp_Containers, [90](#)
- cpp_std_unordered_set_to_py_frozenset< std::complex< double > >
 - Python_Cpp_Containers, [90](#)
- cpp_std_unordered_set_to_py_frozenset< std::string >
 - Python_Cpp_Containers, [90](#)
- cpp_std_unordered_set_to_py_frozenset< std::u16string >
 - Python_Cpp_Containers, [91](#)
- cpp_std_unordered_set_to_py_frozenset< std::u32string >
 - Python_Cpp_Containers, [91](#)
- cpp_std_unordered_set_to_py_frozenset< std::vector< char > >
 - Python_Cpp_Containers, [91](#)
- cpp_std_unordered_set_to_py_set
 - Python_Cpp_Containers, [92](#)
- cpp_std_unordered_set_to_py_set< bool >
 - Python_Cpp_Containers, [92](#)
- cpp_std_unordered_set_to_py_set< double >
 - Python_Cpp_Containers, [93](#)
- cpp_std_unordered_set_to_py_set< long >
 - Python_Cpp_Containers, [93](#)
- cpp_std_unordered_set_to_py_set< std::complex< double > >
 - Python_Cpp_Containers, [93](#)
- cpp_std_unordered_set_to_py_set< std::string >
 - Python_Cpp_Containers, [94](#)
- cpp_std_unordered_set_to_py_set< std::u16string >
 - Python_Cpp_Containers, [94](#)
- cpp_std_unordered_set_to_py_set< std::u32string >
 - Python_Cpp_Containers, [95](#)
- cpp_std_unordered_set_to_py_set< std::vector< char > >
 - Python_Cpp_Containers, [95](#)
- cpp_string_to_py_bytearray
 - Python_Cpp_Containers, [95](#)
- cpp_string_to_py_unicode8
 - Python_Cpp_Containers, [95](#)
- CPP_TYPE_TO_FUNCS
 - src.py.code_gen, [200](#)
- CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINER
 - src.py.code_gen, [200](#)
- cpp_u16string_to_py_unicode16
 - Python_Cpp_Containers, [96](#)
- cpp_u32string_to_py_unicode32
 - Python_Cpp_Containers, [97](#)
- CPP_UNARY_FUNCTION_TO_PY_BASE_DECL
 - src.py.code_gen, [201](#)
- CPP_UNARY_FUNCTION_TO_PY_DECL
 - src.py.code_gen, [201](#)
- CPP_UNARY_FUNCTION_TO_PY_DEFN
 - src.py.code_gen, [201](#)
- cpp_vector_char_to_py_bytearray
 - Python_Cpp_Containers, [97](#)
- cpp_vector_char_to_py_bytes
 - Python_Cpp_Containers, [98](#)
- CppCustomObject, [209](#)
 - CppCustomObject, [210](#)
 - first, [210](#)
 - last, [210](#)
 - name, [210](#)
 - number, [210](#)
- cPyCppContainers.cpp
 - cPyCppContainersMethods, [407](#)
 - cPyCppContainersmodule, [407](#)
 - dict_inc, [392](#)
 - list_x2, [392](#)
 - new_bytes, [392](#)
 - new_dict, [393](#)
 - new_dict_debug, [393](#)
 - new_dict_debug_float_float, [393](#)
 - new_dict_debug_int_int, [394](#)
 - new_dict_from_std_map_bytes_bytes, [394](#)
 - new_dict_from_std_map_complex_complex, [394](#)
 - new_dict_from_std_map_float_float, [394](#)
 - new_dict_from_std_map_int_int, [394](#)
 - new_dict_from_std_map_int_str, [394](#)
 - new_dict_from_std_map_int_str16, [394](#)
 - new_dict_from_std_map_int_str32, [395](#)
 - new_dict_from_std_map_str16_str16, [395](#)
 - new_dict_from_std_map_str32_str32, [395](#)
 - new_dict_from_std_map_str_str, [395](#)
 - new_dict_from_std_unordered_map_bytes_bytes, [395](#)
 - new_dict_from_std_unordered_map_complex_complex, [395](#)
 - new_dict_from_std_unordered_map_float_float, [396](#)
 - new_dict_from_std_unordered_map_int_int, [396](#)
 - new_dict_from_std_unordered_map_int_str, [396](#)
 - new_dict_from_std_unordered_map_int_str16, [396](#)
 - new_dict_from_std_unordered_map_int_str32, [396](#)
 - new_dict_from_std_unordered_map_str16_str16, [396](#)
 - new_dict_from_std_unordered_map_str32_str32, [397](#)
 - new_dict_from_std_unordered_map_str_str, [397](#)
 - new_frozenset, [397](#)

- new_frozenset_bytes, 397
- new_frozenset_complex, 398
- new_frozenset_float, 398
- new_frozenset_int, 398
- new_frozenset_str, 399
- new_list, 399
- new_list_list_bool, 400
- new_list_list_bytes, 400
- new_list_list_complex, 400
- new_list_list_float, 400
- new_list_list_int, 400
- new_list_list_str, 400
- new_list_list_str16, 401
- new_list_list_str32, 401
- new_list_vector_bool, 401
- new_list_vector_bytes, 401
- new_list_vector_complex, 401
- new_list_vector_float, 402
- new_list_vector_int, 402
- new_list_vector_str, 402
- new_list_vector_str16, 402
- new_list_vector_str32, 402
- new_set, 402
- new_set_bytes, 403
- new_set_complex, 403
- new_set_float, 403
- new_set_int, 404
- new_set_str, 404
- new_set_str16, 405
- new_set_str32, 405
- new_str, 405
- new_str16, 405
- new_str32, 406
- PY_SSIZE_T_CLEAN, 391
- PyInit_cPyCppContainers, 406
- reverse_vector, 406
- SINGLE_ARGUMENT_METHOD, 391
- tuple_reverse, 406
- vector_double_x2, 407
- cPyCppContainersMethods
 - cPyCppContainers.cpp, 407
- cPyCppContainersmodule
 - cPyCppContainers.cpp, 407
- cUserDefined.cpp
 - cpp_custom_object_to_py_custom_object, 409
 - cUserDefinedMethods, 411
 - cUserDefinedmodule, 412
 - Custom_dealloc, 409
 - Custom_getfirst, 409
 - Custom_getlast, 410
 - Custom_getsetters, 412
 - Custom_init, 410
 - Custom_members, 412
 - Custom_methods, 412
 - Custom_name, 410
 - Custom_new, 410
 - Custom_setfirst, 410
 - Custom_setlast, 410
- CustomType, 413
 - py_custom_object_check, 411
 - py_custom_object_to_cpp_custom_object, 411
 - PY_SSIZE_T_CLEAN, 409
 - PyInit_cUserDefined, 411
 - reverse_dict_names, 411
 - reverse_list_names, 411
- cUserDefinedMethods
 - cUserDefined.cpp, 411
- cUserDefinedmodule
 - cUserDefined.cpp, 412
- Custom_dealloc
 - cUserDefined.cpp, 409
- Custom_getfirst
 - cUserDefined.cpp, 409
- Custom_getlast
 - cUserDefined.cpp, 410
- Custom_getsetters
 - cUserDefined.cpp, 412
- Custom_init
 - cUserDefined.cpp, 410
- Custom_members
 - cUserDefined.cpp, 412
- Custom_methods
 - cUserDefined.cpp, 412
- Custom_name
 - cUserDefined.cpp, 410
- Custom_new
 - cUserDefined.cpp, 410
- Custom_setfirst
 - cUserDefined.cpp, 410
- Custom_setlast
 - cUserDefined.cpp, 410
- CustomObject, 211
 - first, 211
 - last, 211
 - number, 211
- CustomType
 - cUserDefined.cpp, 413
- declarations
 - src.py.code_gen, 196
- definitions
 - src.py.code_gen, 197
- defn_name_from_decl_name
 - src.py.code_gen, 197
- dict_inc
 - cPyCppContainers.cpp, 392
- dict_map_declarations
 - src.py.code_gen, 197
- dict_map_definitions
 - src.py.code_gen, 197
- documentation
 - src.py.code_gen_documentation, 204
- doubles_cmp
 - test_internal.cpp, 345
- doxygen_cpp_to_python_dict_base_class
 - src.py.code_gen_documentation, 204
- doxygen_cpp_to_python_dict_instantiation

- src.py.code_gen_documentation, 205
- doxygen_cpp_to_python_unary_base_class
 - src.py.code_gen_documentation, 205
- doxygen_cpp_to_python_unary_instantiation
 - src.py.code_gen_documentation, 205
- doxygen_python_dict_to_cpp_base_class
 - src.py.code_gen_documentation, 206
- doxygen_python_dict_to_cpp_instantiation
 - src.py.code_gen_documentation, 206
- doxygen_python_to_cpp_unary_base_class
 - src.py.code_gen_documentation, 207
- doxygen_python_to_cpp_unary_instantiation
 - src.py.code_gen_documentation, 207
- dump_header
 - TestResultS, 228
- dump_tail
 - TestResultS, 228
- dump_tests
 - TestResultS, 228
- ErrorReturnValue
 - Python_Cpp_Containers, 29
- ExecClock, 212
 - ExecClock, 212
 - seconds, 212
 - thiResDouble, 212
- execTime
 - TestResult, 224
- execTimeAdd
 - TestResult, 224
- execTimeMax
 - TestResult, 224
- execTimeMin
 - TestResult, 225
- execTimeStdDev
 - TestResult, 225
- explore_hash_reserve
 - main.cpp, 415
- FAIL_CONTAINER_KEY_WRONG_TYPE
 - Python_Cpp_Containers, 29
- FAIL_CONTAINER_MEMBER_WRONG_TYPE
 - Python_Cpp_Containers, 29
- FAIL_CONTAINER_VALUE_WRONG_TYPE
 - Python_Cpp_Containers, 29
- FAIL_CONTAINER_WRONG_TYPE
 - Python_Cpp_Containers, 29
- failed
 - TestResult, 225
 - TestResultS, 228
- failure
 - SubTestCount, 219
- first
 - CppCustomObject, 210
 - CustomObject, 211
- generic_cpp_std_list_like_to_py_list
 - Python_Cpp_Containers, 98
- generic_cpp_std_list_like_to_py_list_like
 - Python_Cpp_Containers, 99
- generic_cpp_std_list_like_to_py_tuple
 - Python_Cpp_Containers, 99, 100
- generic_cpp_std_list_to_py_list
 - Python_Cpp_Containers, 100
- generic_cpp_std_list_to_py_tuple
 - Python_Cpp_Containers, 101
- generic_cpp_std_map_like_to_py_dict
 - Python_Cpp_Containers, 101
- generic_cpp_std_unordered_set_to_py_frozenset
 - Python_Cpp_Containers, 102
- generic_cpp_std_unordered_set_to_py_set
 - Python_Cpp_Containers, 102
- generic_cpp_std_unordered_set_to_py_set_or_frozenset
 - Python_Cpp_Containers, 103
- generic_cpp_std_vector_to_py_list
 - Python_Cpp_Containers, 103
- generic_cpp_std_vector_to_py_tuple
 - Python_Cpp_Containers, 103
- generic_py_dict_to_cpp_std_map_like
 - Python_Cpp_Containers, 103
- generic_py_frozenset_to_cpp_std_unordered_set
 - Python_Cpp_Containers, 104
- generic_py_list_to_cpp_std_list
 - Python_Cpp_Containers, 105
- generic_py_list_to_cpp_std_list_like
 - Python_Cpp_Containers, 105
- generic_py_list_to_cpp_std_vector
 - Python_Cpp_Containers, 106
- generic_py_set_or_frozenset_to_cpp_std_unordered_set
 - Python_Cpp_Containers, 106
- generic_py_set_to_cpp_std_unordered_set
 - Python_Cpp_Containers, 107
- generic_py_tuple_to_cpp_std_list
 - Python_Cpp_Containers, 107
- generic_py_tuple_to_cpp_std_list_like
 - Python_Cpp_Containers, 107, 108
- generic_py_tuple_to_cpp_std_vector
 - Python_Cpp_Containers, 109
- generic_py_unary_to_cpp_std_list_like
 - Python_Cpp_Containers, 109
- get_codegen_please_no_edit_warning
 - src.py.code_gen_documentation, 207
- get_codegen_please_no_edit_warning_context
 - src.py.code_gen_documentation, 208
- get_rss.cpp
 - getCurrentRSS, 232
 - getCurrentRSS_alterate, 232
 - getCurrentRSS_alterateMb, 232
 - getCurrentRSSMb, 233
 - getPeakRSS, 233
 - getPeakRSSMb, 233
 - MB_PRECISION, 233
 - MB_WIDTH, 233
 - MEGABYTES, 233
 - operator<<, 233
 - RSS_SNAPSHOT_REPORT_PAGES, 232
- get_rss.h

- getCurrentRSS, [234](#)
- getCurrentRSS_alterate, [234](#)
- getPeakRSS, [235](#)
- MEGABYTES, [235](#)
- operator<<, [235](#)
- getCurrentRSS
 - get_rss.cpp, [232](#)
 - get_rss.h, [234](#)
- getCurrentRSS_alterate
 - get_rss.cpp, [232](#)
 - get_rss.h, [234](#)
- getCurrentRSS_alterateMb
 - get_rss.cpp, [232](#)
- getCurrentRSSMb
 - get_rss.cpp, [233](#)
- getPeakRSS
 - get_rss.cpp, [233](#)
 - get_rss.h, [235](#)
- getPeakRSSMb
 - get_rss.cpp, [233](#)
- hasExecTimeStdDev
 - TestResult, [225](#)
- INC_SIZE_OF_CONTAINER_MULTIPLE
 - test_performance.cpp, [388](#)
- INC_STRING_LENGTH_MULTIPLE
 - test_performance.cpp, [388](#)
- last
 - CppCustomObject, [210](#)
 - CustomObject, [211](#)
- LIMIT_SIZE_OF_CONTAINER
 - test_performance.cpp, [388](#)
- LIMIT_SIZE_OF_CONTAINER_DICT
 - test_performance.cpp, [388](#)
- LIMIT_STRING_LENGTH
 - test_performance.cpp, [388](#)
- list_x2
 - cPyCppContainers.cpp, [392](#)
- logger
 - src.py.code_gen, [201](#)
- m_failure
 - SubTestCount, [220](#)
- m_name
 - RSSSnapshot, [217](#)
- m_rss_initial
 - RSSSnapshot, [217](#)
- m_rss_peak_initial
 - RSSSnapshot, [217](#)
- m_test_count
 - SubTestCount, [220](#)
- main
 - main.cpp, [415](#)
 - src.py.code_gen, [197](#)
- main.cpp
 - explore_hash_reserve, [415](#)
 - main, [415](#)
 - test_all, [415](#)
 - TEST_FUNCTIONAL_ALL, [414](#)
 - TEST_INTERNAL_ALL, [415](#)
 - TEST_MEMORY_ALL, [415](#)
 - TEST_PERFORMANCE_ALL, [415](#)
- MB_PRECISION
 - get_rss.cpp, [233](#)
- MB_WIDTH
 - get_rss.cpp, [233](#)
- MEGABYTES
 - get_rss.cpp, [233](#)
 - get_rss.h, [235](#)
- MIN_SIZE_OF_CONTAINER
 - test_performance.cpp, [388](#)
- MIN_STRING_LENGTH_HASHABLE
 - test_performance.cpp, [388](#)
- MIN_STRING_LENGTH_NON_HASHABLE
 - test_performance.cpp, [389](#)
- name
 - CppCustomObject, [210](#)
 - RSSSnapshot, [215](#)
 - TestResult, [225](#)
- new_bytes
 - cPyCppContainers.cpp, [392](#)
- new_dict
 - cPyCppContainers.cpp, [393](#)
- new_dict_debug
 - cPyCppContainers.cpp, [393](#)
- new_dict_debug_float_float
 - cPyCppContainers.cpp, [393](#)
- new_dict_debug_int_int
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_bytes_bytes
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_complex_complex
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_float_float
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_int_int
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_int_str
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_int_str16
 - cPyCppContainers.cpp, [394](#)
- new_dict_from_std_map_int_str32
 - cPyCppContainers.cpp, [395](#)
- new_dict_from_std_map_str16_str16
 - cPyCppContainers.cpp, [395](#)
- new_dict_from_std_map_str32_str32
 - cPyCppContainers.cpp, [395](#)
- new_dict_from_std_map_str_str
 - cPyCppContainers.cpp, [395](#)
- new_dict_from_std_unordered_map_bytes_bytes
 - cPyCppContainers.cpp, [395](#)
- new_dict_from_std_unordered_map_complex_complex
 - cPyCppContainers.cpp, [395](#)
- new_dict_from_std_unordered_map_float_float
 - cPyCppContainers.cpp, [396](#)

- `new_dict_from_std_unordered_map_int_int`
 `cPyCppContainers.cpp`, 396
- `new_dict_from_std_unordered_map_int_str`
 `cPyCppContainers.cpp`, 396
- `new_dict_from_std_unordered_map_int_str16`
 `cPyCppContainers.cpp`, 396
- `new_dict_from_std_unordered_map_int_str32`
 `cPyCppContainers.cpp`, 396
- `new_dict_from_std_unordered_map_str16_str16`
 `cPyCppContainers.cpp`, 396
- `new_dict_from_std_unordered_map_str32_str32`
 `cPyCppContainers.cpp`, 397
- `new_dict_from_std_unordered_map_str_str`
 `cPyCppContainers.cpp`, 397
- `new_frozenset`
 `cPyCppContainers.cpp`, 397
- `new_frozenset_bytes`
 `cPyCppContainers.cpp`, 397
- `new_frozenset_complex`
 `cPyCppContainers.cpp`, 398
- `new_frozenset_float`
 `cPyCppContainers.cpp`, 398
- `new_frozenset_int`
 `cPyCppContainers.cpp`, 398
- `new_frozenset_str`
 `cPyCppContainers.cpp`, 399
- `new_list`
 `cPyCppContainers.cpp`, 399
- `new_list_list_bool`
 `cPyCppContainers.cpp`, 400
- `new_list_list_bytes`
 `cPyCppContainers.cpp`, 400
- `new_list_list_complex`
 `cPyCppContainers.cpp`, 400
- `new_list_list_float`
 `cPyCppContainers.cpp`, 400
- `new_list_list_int`
 `cPyCppContainers.cpp`, 400
- `new_list_list_str`
 `cPyCppContainers.cpp`, 400
- `new_list_list_str16`
 `cPyCppContainers.cpp`, 401
- `new_list_list_str32`
 `cPyCppContainers.cpp`, 401
- `new_list_vector_bool`
 `cPyCppContainers.cpp`, 401
- `new_list_vector_bytes`
 `cPyCppContainers.cpp`, 401
- `new_list_vector_complex`
 `cPyCppContainers.cpp`, 401
- `new_list_vector_float`
 `cPyCppContainers.cpp`, 402
- `new_list_vector_int`
 `cPyCppContainers.cpp`, 402
- `new_list_vector_str`
 `cPyCppContainers.cpp`, 402
- `new_list_vector_str16`
 `cPyCppContainers.cpp`, 402
- `new_list_vector_str32`
 `cPyCppContainers.cpp`, 402
- `new_py_dict_bytes`
 `test_common.cpp`, 282
 `test_common.h`, 316
- `new_py_dict_string`
 `test_common.cpp`, 282
 `test_common.h`, 317
- `new_py_dict_string16`
 `test_common.cpp`, 282
 `test_common.h`, 317
- `new_py_dict_string32`
 `test_common.cpp`, 283
 `test_common.h`, 318
- `new_py_list_bytes`
 `test_common.cpp`, 283
 `test_common.h`, 318
- `new_py_list_string`
 `test_common.cpp`, 284
 `test_common.h`, 318
- `new_py_list_string16`
 `test_common.cpp`, 284
 `test_common.h`, 319
- `new_py_list_string32`
 `test_common.cpp`, 284
 `test_common.h`, 319
- `new_py_set_bytes`
 `test_common.cpp`, 285
 `test_common.h`, 319
- `new_py_set_string`
 `test_common.cpp`, 285
 `test_common.h`, 320
- `new_py_set_u16string`
 `test_common.cpp`, 285
 `test_common.h`, 320
- `new_py_set_u32string`
 `test_common.cpp`, 286
 `test_common.h`, 321
- `new_py_tuple_bytes`
 `test_common.cpp`, 286
 `test_common.h`, 321
- `new_py_tuple_string`
 `test_common.cpp`, 287
 `test_common.h`, 321
- `new_py_tuple_string16`
 `test_common.cpp`, 287
 `test_common.h`, 322
- `new_py_tuple_string32`
 `test_common.cpp`, 287
 `test_common.h`, 322
- `new_set`
 `cPyCppContainers.cpp`, 402
- `new_set_bytes`
 `cPyCppContainers.cpp`, 403
- `new_set_complex`
 `cPyCppContainers.cpp`, 403
- `new_set_float`
 `cPyCppContainers.cpp`, 403

- new_set_int
 - cPyCppContainers.cpp, 404
- new_set_str
 - cPyCppContainers.cpp, 404
- new_set_str16
 - cPyCppContainers.cpp, 405
- new_set_str32
 - cPyCppContainers.cpp, 405
- new_str
 - cPyCppContainers.cpp, 405
- new_str16
 - cPyCppContainers.cpp, 405
- new_str32
 - cPyCppContainers.cpp, 406
- number
 - CppCustomObject, 210
 - CustomObject, 211
- numScaleValues
 - TestResult, 225
- numTests
 - TestResult, 226
- operator<<
 - get_rss.cpp, 233
 - get_rss.h, 235
 - TestFramework.cpp, 237, 239
 - TestFramework.h, 242
- operator()
 - std::hash< std::complex< double > >, 213
 - std::hash< std::vector< char > >, 214
 - std::less< std::complex< T > >, 214
- operator=
 - TestResult, 226
- PROJECT_VERSION
 - src.py.code_gen, 201
- push_back
 - TestResultS, 228
- py_bool_check
 - Python_Cpp_Containers, 109
- py_bool_to_cpp_bool
 - Python_Cpp_Containers, 109
- py_bytearray_check
 - Python_Cpp_Containers, 110
- py_bytearray_to_cpp_string
 - Python_Cpp_Containers, 110
- py_bytearray_to_cpp_vector_char
 - Python_Cpp_Containers, 110
- py_bytes_check
 - Python_Cpp_Containers, 111
- py_bytes_to_cpp_vector_char
 - Python_Cpp_Containers, 111
- py_complex_check
 - Python_Cpp_Containers, 111
- py_complex_to_cpp_complex
 - Python_Cpp_Containers, 112
- py_custom_object_check
 - cUserDefined.cpp, 411
- py_custom_object_to_cpp_custom_object
 - cUserDefined.cpp, 411
- py_dict_to_cpp_std_map_like
 - Python_Cpp_Containers, 112, 113
- py_dict_to_cpp_std_map_like< std::map, bool, bool >
 - Python_Cpp_Containers, 113
- py_dict_to_cpp_std_map_like< std::map, bool, double >
 - Python_Cpp_Containers, 113
- py_dict_to_cpp_std_map_like< std::map, bool, long >
 - Python_Cpp_Containers, 114
- py_dict_to_cpp_std_map_like< std::map, bool, std::complex< double > >
 - Python_Cpp_Containers, 114
- py_dict_to_cpp_std_map_like< std::map, bool, std::string >
 - Python_Cpp_Containers, 115
- py_dict_to_cpp_std_map_like< std::map, bool, std::u16string >
 - Python_Cpp_Containers, 115
- py_dict_to_cpp_std_map_like< std::map, bool, std::u32string >
 - Python_Cpp_Containers, 116
- py_dict_to_cpp_std_map_like< std::map, bool, std::vector< char > >
 - Python_Cpp_Containers, 116
- py_dict_to_cpp_std_map_like< std::map, double, bool >
 - Python_Cpp_Containers, 116
- py_dict_to_cpp_std_map_like< std::map, double, double >
 - Python_Cpp_Containers, 117
- py_dict_to_cpp_std_map_like< std::map, double, long >
 - Python_Cpp_Containers, 117
- py_dict_to_cpp_std_map_like< std::map, double, std::complex< double > >
 - Python_Cpp_Containers, 118
- py_dict_to_cpp_std_map_like< std::map, double, std::string >
 - Python_Cpp_Containers, 118
- py_dict_to_cpp_std_map_like< std::map, double, std::u16string >
 - Python_Cpp_Containers, 119
- py_dict_to_cpp_std_map_like< std::map, double, std::u32string >
 - Python_Cpp_Containers, 119
- py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > >
 - Python_Cpp_Containers, 119
- py_dict_to_cpp_std_map_like< std::map, long, bool >
 - Python_Cpp_Containers, 120
- py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject >
 - Python_Cpp_Containers, 120
- py_dict_to_cpp_std_map_like< std::map, long, double >
 - Python_Cpp_Containers, 120
- py_dict_to_cpp_std_map_like< std::map, long, long >
 - Python_Cpp_Containers, 120

- Python_Cpp_Containers, [121](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::complex< double > >
 - Python_Cpp_Containers, [121](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::string >
 - Python_Cpp_Containers, [122](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::u16string >
 - Python_Cpp_Containers, [122](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::u32string >
 - Python_Cpp_Containers, [123](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > >
 - Python_Cpp_Containers, [123](#)
- py_dict_to_cpp_std_map_like< std::map, std::complex< double >, bool >
 - Python_Cpp_Containers, [123](#)
- py_dict_to_cpp_std_map_like< std::map, std::complex< double >, double >
 - Python_Cpp_Containers, [124](#)
- py_dict_to_cpp_std_map_like< std::map, std::complex< double >, long >
 - Python_Cpp_Containers, [124](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, bool >
 - Python_Cpp_Containers, [125](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, double >
 - Python_Cpp_Containers, [125](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, long >
 - Python_Cpp_Containers, [126](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::complex< double > >
 - Python_Cpp_Containers, [126](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::string >
 - Python_Cpp_Containers, [127](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string >
 - Python_Cpp_Containers, [127](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::u32string >
 - Python_Cpp_Containers, [127](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::vector< char > >
 - Python_Cpp_Containers, [128](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, bool >
 - Python_Cpp_Containers, [128](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, double >
 - Python_Cpp_Containers, [130](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, long >
 - Python_Cpp_Containers, [130](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, std::complex< double > >
 - Python_Cpp_Containers, [131](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, std::string >
 - Python_Cpp_Containers, [131](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u16string >
 - Python_Cpp_Containers, [131](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, std::u32string >
 - Python_Cpp_Containers, [132](#)
- py_dict_to_cpp_std_map_like< std::map, std::u16string, std::vector< char > >
 - Python_Cpp_Containers, [132](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, bool >
 - Python_Cpp_Containers, [133](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, double >
 - Python_Cpp_Containers, [133](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, long >
 - Python_Cpp_Containers, [134](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, std::complex< double > >
 - Python_Cpp_Containers, [134](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, std::string >
 - Python_Cpp_Containers, [135](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u16string >
 - Python_Cpp_Containers, [135](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, std::u32string >
 - Python_Cpp_Containers, [135](#)
- py_dict_to_cpp_std_map_like< std::map, std::u32string, std::vector< char > >
 - Python_Cpp_Containers, [136](#)
- py_dict_to_cpp_std_map_like< std::map, std::vector< char >, bool >
 - Python_Cpp_Containers, [136](#)
- py_dict_to_cpp_std_map_like< std::map, std::vector< char >, double >
 - Python_Cpp_Containers, [137](#)
- py_dict_to_cpp_std_map_like< std::map, std::vector< char >, long >
 - Python_Cpp_Containers, [137](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, bool >
 - Python_Cpp_Containers, [138](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, double >
 - Python_Cpp_Containers, [138](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, long >
 - Python_Cpp_Containers, [139](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,

- bool, std::complex< double > >
 - Python_Cpp_Containers, [139](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::string >
 - Python_Cpp_Containers, [139](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u16string >
 - Python_Cpp_Containers, [140](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::u32string >
 - Python_Cpp_Containers, [140](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, bool, std::vector< char > >
 - Python_Cpp_Containers, [141](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, bool >
 - Python_Cpp_Containers, [141](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, double >
 - Python_Cpp_Containers, [142](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, long >
 - Python_Cpp_Containers, [142](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, std::complex< double > >
 - Python_Cpp_Containers, [143](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, std::string >
 - Python_Cpp_Containers, [143](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u16string >
 - Python_Cpp_Containers, [143](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, std::u32string >
 - Python_Cpp_Containers, [144](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, double, std::vector< char > >
 - Python_Cpp_Containers, [144](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, bool >
 - Python_Cpp_Containers, [145](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, double >
 - Python_Cpp_Containers, [145](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, long >
 - Python_Cpp_Containers, [146](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex< double > >
 - Python_Cpp_Containers, [146](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, std::string >
 - Python_Cpp_Containers, [147](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u16string >
 - Python_Cpp_Containers, [147](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, std::u32string >
 - Python_Cpp_Containers, [147](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, std::vector< char > >
 - Python_Cpp_Containers, [147](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, long, std::complex< double >, bool >
 - Python_Cpp_Containers, [148](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, double >
 - Python_Cpp_Containers, [149](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::complex< double >, long >
 - Python_Cpp_Containers, [149](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, bool >
 - Python_Cpp_Containers, [150](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, double >
 - Python_Cpp_Containers, [150](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, long >
 - Python_Cpp_Containers, [151](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::complex< double > >
 - Python_Cpp_Containers, [151](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::string >
 - Python_Cpp_Containers, [151](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u16string >
 - Python_Cpp_Containers, [152](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::u32string >
 - Python_Cpp_Containers, [152](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::string, std::vector< char > >
 - Python_Cpp_Containers, [153](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, bool >
 - Python_Cpp_Containers, [153](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, double >
 - Python_Cpp_Containers, [154](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, long >
 - Python_Cpp_Containers, [154](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::complex< double > >
 - Python_Cpp_Containers, [155](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::string >
 - Python_Cpp_Containers, [155](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u16string >
 - Python_Cpp_Containers, [155](#)
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u32string >
 - Python_Cpp_Containers, [156](#)

- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u16string, std::vector< char > >
Python_Cpp_Containers, [156](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, bool >
Python_Cpp_Containers, [157](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, double >
Python_Cpp_Containers, [157](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, long >
Python_Cpp_Containers, [158](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, std::complex< double > >
Python_Cpp_Containers, [158](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, std::string >
Python_Cpp_Containers, [159](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, std::u16string >
Python_Cpp_Containers, [159](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, std::u32string >
Python_Cpp_Containers, [159](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::u32string, std::vector< char > >
Python_Cpp_Containers, [160](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::vector< char >, bool >
Python_Cpp_Containers, [160](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::vector< char >, double >
Python_Cpp_Containers, [161](#)
- py_dict_to_cpp_std_map_like< std::unordered_map,
std::vector< char >, long >
Python_Cpp_Containers, [161](#)
- PY_ERR_ON_ENTRY_RETURN_CODE
test_common.h, [339](#)
- PY_ERR_ON_EXIT_RETURN_CODE
test_common.h, [339](#)
- py_float_check
Python_Cpp_Containers, [162](#)
- py_float_to_cpp_double
Python_Cpp_Containers, [162](#)
- py_frozenset_check
Python_Cpp_Containers, [162](#)
- py_frozenset_to_cpp_std_unordered_set
Python_Cpp_Containers, [163](#)
- py_frozenset_to_cpp_std_unordered_set< bool >
Python_Cpp_Containers, [163](#)
- py_frozenset_to_cpp_std_unordered_set< double >
Python_Cpp_Containers, [164](#)
- py_frozenset_to_cpp_std_unordered_set< long >
Python_Cpp_Containers, [164](#)
- py_frozenset_to_cpp_std_unordered_set< std::complex<
double > >
Python_Cpp_Containers, [165](#)
- py_frozenset_to_cpp_std_unordered_set< std::string >
Python_Cpp_Containers, [165](#)
- py_frozenset_to_cpp_std_unordered_set< std::u16string
>
Python_Cpp_Containers, [165](#)
- py_frozenset_to_cpp_std_unordered_set< std::u32string
>
Python_Cpp_Containers, [166](#)
- py_frozenset_to_cpp_std_unordered_set< std::vector<
char > >
Python_Cpp_Containers, [166](#)
- py_list_check
Python_Cpp_Containers, [167](#)
- py_list_get
Python_Cpp_Containers, [167](#)
- py_list_len
Python_Cpp_Containers, [167](#)
- py_list_new
Python_Cpp_Containers, [168](#)
- py_list_set
Python_Cpp_Containers, [168](#)
- py_list_to_cpp_std_list_like
Python_Cpp_Containers, [169](#)
- py_list_to_cpp_std_list_like< bool >
Python_Cpp_Containers, [170](#)
- py_list_to_cpp_std_list_like< CppCustomObject >
Python_Cpp_Containers, [170](#)
- py_list_to_cpp_std_list_like< double >
Python_Cpp_Containers, [171](#)
- py_list_to_cpp_std_list_like< long >
Python_Cpp_Containers, [171](#), [172](#)
- py_list_to_cpp_std_list_like< std::complex< double >
>
Python_Cpp_Containers, [172](#), [173](#)
- py_list_to_cpp_std_list_like< std::string >
Python_Cpp_Containers, [173](#)
- py_list_to_cpp_std_list_like< std::u16string >
Python_Cpp_Containers, [174](#)
- py_list_to_cpp_std_list_like< std::u32string >
Python_Cpp_Containers, [175](#)
- py_list_to_cpp_std_list_like< std::vector< char > >
Python_Cpp_Containers, [175](#), [176](#)
- py_long_check
Python_Cpp_Containers, [176](#)
- py_long_to_cpp_long
Python_Cpp_Containers, [177](#)
- py_set_check
Python_Cpp_Containers, [177](#)
- py_set_to_cpp_std_unordered_set
Python_Cpp_Containers, [177](#)
- py_set_to_cpp_std_unordered_set< bool >
Python_Cpp_Containers, [178](#)
- py_set_to_cpp_std_unordered_set< double >
Python_Cpp_Containers, [178](#)
- py_set_to_cpp_std_unordered_set< long >
Python_Cpp_Containers, [179](#)
- py_set_to_cpp_std_unordered_set< std::complex<
double > >
Python_Cpp_Containers, [179](#)

- py_set_to_cpp_std_unordered_set< std::string >
 - Python_Cpp_Containers, 179
- py_set_to_cpp_std_unordered_set< std::u16string >
 - Python_Cpp_Containers, 180
- py_set_to_cpp_std_unordered_set< std::u32string >
 - Python_Cpp_Containers, 180
- py_set_to_cpp_std_unordered_set< std::vector< char > >
 - Python_Cpp_Containers, 181
- PY_SSIZE_T_CLEAN
 - cPyCppContainers.cpp, 391
 - cUserDefined.cpp, 409
- PY_TO_CPP_UNARY_FUNCTION_BASE_DECL
 - src.py.code_gen, 202
- PY_TO_CPP_UNARY_FUNCTION_DECL
 - src.py.code_gen, 202
- PY_TO_CPP_UNARY_FUNCTION_DEFN
 - src.py.code_gen, 202
- py_tuple_check
 - Python_Cpp_Containers, 181
- py_tuple_get
 - Python_Cpp_Containers, 181
- py_tuple_len
 - Python_Cpp_Containers, 182
- py_tuple_new
 - Python_Cpp_Containers, 182
- py_tuple_set
 - Python_Cpp_Containers, 183
- py_tuple_to_cpp_std_list_like
 - Python_Cpp_Containers, 183, 184
- py_tuple_to_cpp_std_list_like< bool >
 - Python_Cpp_Containers, 184, 185
- py_tuple_to_cpp_std_list_like< double >
 - Python_Cpp_Containers, 185, 186
- py_tuple_to_cpp_std_list_like< long >
 - Python_Cpp_Containers, 186
- py_tuple_to_cpp_std_list_like< std::complex< double > >
 - Python_Cpp_Containers, 187
- py_tuple_to_cpp_std_list_like< std::string >
 - Python_Cpp_Containers, 188
- py_tuple_to_cpp_std_list_like< std::u16string >
 - Python_Cpp_Containers, 188, 189
- py_tuple_to_cpp_std_list_like< std::u32string >
 - Python_Cpp_Containers, 189, 190
- py_tuple_to_cpp_std_list_like< std::vector< char > >
 - Python_Cpp_Containers, 190
- py_unicode16_check
 - Python_Cpp_Containers, 191
- py_unicode16_to_cpp_u16string
 - Python_Cpp_Containers, 191
- py_unicode32_check
 - Python_Cpp_Containers, 192
- py_unicode32_to_cpp_u32string
 - Python_Cpp_Containers, 192
- py_unicode8_check
 - Python_Cpp_Containers, 192
- py_unicode8_to_cpp_string
 - Python_Cpp_Containers, 193
- PyInit_cPyCppContainers
 - cPyCppContainers.cpp, 406
- PyInit_cUserDefined
 - cUserDefined.cpp, 411
- python_convert.h
 - PYTHON_CPP_CONTAINERS_VERSION, 271
- Python_Cpp_Containers, 15
 - cpp_bool_to_py_bool, 29
 - cpp_complex_to_py_complex, 30
 - cpp_double_to_py_float, 30
 - cpp_long_to_py_long, 30
 - cpp_std_list_like_to_py_list, 31
 - cpp_std_list_like_to_py_list< bool >, 31, 32
 - cpp_std_list_like_to_py_list< CppCustomObject >, 32
 - cpp_std_list_like_to_py_list< double >, 32, 33
 - cpp_std_list_like_to_py_list< long >, 33
 - cpp_std_list_like_to_py_list< std::complex< double > >, 34
 - cpp_std_list_like_to_py_list< std::string >, 35
 - cpp_std_list_like_to_py_list< std::u16string >, 35, 36
 - cpp_std_list_like_to_py_list< std::u32string >, 36
 - cpp_std_list_like_to_py_list< std::vector< char > >, 37
 - cpp_std_list_like_to_py_tuple, 38
 - cpp_std_list_like_to_py_tuple< bool >, 38, 39
 - cpp_std_list_like_to_py_tuple< double >, 39, 40
 - cpp_std_list_like_to_py_tuple< long >, 40
 - cpp_std_list_like_to_py_tuple< std::complex< double > >, 41
 - cpp_std_list_like_to_py_tuple< std::string >, 41, 42
 - cpp_std_list_like_to_py_tuple< std::u16string >, 42, 43
 - cpp_std_list_like_to_py_tuple< std::u32string >, 43
 - cpp_std_list_like_to_py_tuple< std::vector< char > >, 44
 - cpp_std_map_like_to_py_dict, 44, 45
 - cpp_std_map_like_to_py_dict< std::map, bool, bool >, 45
 - cpp_std_map_like_to_py_dict< std::map, bool, double >, 46
 - cpp_std_map_like_to_py_dict< std::map, bool, long >, 46
 - cpp_std_map_like_to_py_dict< std::map, bool, std::complex< double > >, 46
 - cpp_std_map_like_to_py_dict< std::map, bool, std::string >, 47
 - cpp_std_map_like_to_py_dict< std::map, bool, std::u16string >, 47
 - cpp_std_map_like_to_py_dict< std::map, bool, std::u32string >, 48
 - cpp_std_map_like_to_py_dict< std::map, bool, std::vector< char > >, 48
 - cpp_std_map_like_to_py_dict< std::map, double,

- bool >, 48
- cpp_std_map_like_to_py_dict< std::map, double, double >, 49
- cpp_std_map_like_to_py_dict< std::map, double, long >, 49
- cpp_std_map_like_to_py_dict< std::map, double, std::complex< double > >, 50
- cpp_std_map_like_to_py_dict< std::map, double, std::string >, 50
- cpp_std_map_like_to_py_dict< std::map, double, std::u16string >, 50
- cpp_std_map_like_to_py_dict< std::map, double, std::u32string >, 51
- cpp_std_map_like_to_py_dict< std::map, double, std::vector< char > >, 51
- cpp_std_map_like_to_py_dict< std::map, long, bool >, 52
- cpp_std_map_like_to_py_dict< std::map, long, CppCustomObject >, 52
- cpp_std_map_like_to_py_dict< std::map, long, double >, 52
- cpp_std_map_like_to_py_dict< std::map, long, long >, 52
- cpp_std_map_like_to_py_dict< std::map, long, std::complex< double > >, 53
- cpp_std_map_like_to_py_dict< std::map, long, std::string >, 53
- cpp_std_map_like_to_py_dict< std::map, long, std::u16string >, 54
- cpp_std_map_like_to_py_dict< std::map, long, std::u32string >, 54
- cpp_std_map_like_to_py_dict< std::map, long, std::vector< char > >, 54
- cpp_std_map_like_to_py_dict< std::map, std::complex< double >, bool >, 55
- cpp_std_map_like_to_py_dict< std::map, std::complex< double >, double >, 55
- cpp_std_map_like_to_py_dict< std::map, std::complex< double >, long >, 56
- cpp_std_map_like_to_py_dict< std::map, std::string, bool >, 56
- cpp_std_map_like_to_py_dict< std::map, std::string, double >, 56
- cpp_std_map_like_to_py_dict< std::map, std::string, long >, 57
- cpp_std_map_like_to_py_dict< std::map, std::string, std::complex< double > >, 57
- cpp_std_map_like_to_py_dict< std::map, std::string, std::string >, 58
- cpp_std_map_like_to_py_dict< std::map, std::string, std::u16string >, 58
- cpp_std_map_like_to_py_dict< std::map, std::string, std::u32string >, 58
- cpp_std_map_like_to_py_dict< std::map, std::string, std::vector< char > >, 59
- cpp_std_map_like_to_py_dict< std::map, std::u16string, bool >, 59
- cpp_std_map_like_to_py_dict< std::map, std::u16string, double >, 60
- cpp_std_map_like_to_py_dict< std::map, std::u16string, long >, 60
- cpp_std_map_like_to_py_dict< std::map, std::u16string, std::complex< double > >, 60
- cpp_std_map_like_to_py_dict< std::map, std::u16string, std::string >, 61
- cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u16string >, 61
- cpp_std_map_like_to_py_dict< std::map, std::u16string, std::u32string >, 62
- cpp_std_map_like_to_py_dict< std::map, std::u16string, std::vector< char > >, 62
- cpp_std_map_like_to_py_dict< std::map, std::u32string, bool >, 62
- cpp_std_map_like_to_py_dict< std::map, std::u32string, double >, 63
- cpp_std_map_like_to_py_dict< std::map, std::u32string, long >, 63
- cpp_std_map_like_to_py_dict< std::map, std::u32string, std::complex< double > >, 64
- cpp_std_map_like_to_py_dict< std::map, std::u32string, std::string >, 64
- cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u16string >, 64
- cpp_std_map_like_to_py_dict< std::map, std::u32string, std::u32string >, 65
- cpp_std_map_like_to_py_dict< std::map, std::u32string, std::vector< char > >, 65
- cpp_std_map_like_to_py_dict< std::map, std::vector< char >, bool >, 66
- cpp_std_map_like_to_py_dict< std::map, std::vector< char >, double >, 66
- cpp_std_map_like_to_py_dict< std::map, std::vector< char >, long >, 66
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, bool >, 67
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, double >, 67
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, long >, 68
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::complex< double > >, 68
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::string >, 68
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u16string >, 69
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::u32string >, 69
- cpp_std_map_like_to_py_dict< std::unordered_map, bool, std::vector< char > >, 70
- cpp_std_map_like_to_py_dict< std::unordered_map, double, bool >, 70
- cpp_std_map_like_to_py_dict< std::unordered_map, double, double >, 70
- cpp_std_map_like_to_py_dict< std::unordered_map, double, long >, 71
- cpp_std_map_like_to_py_dict< std::unordered_map,

- double, std::complex< double > >, 71
- cpp_std_map_like_to_py_dict< std::unordered_map, double, std::string >, 72
- cpp_std_map_like_to_py_dict< std::unordered_map, double, std::u16string >, 72
- cpp_std_map_like_to_py_dict< std::unordered_map, double, std::u32string >, 72
- cpp_std_map_like_to_py_dict< std::unordered_map, double, std::vector< char > >, 73
- cpp_std_map_like_to_py_dict< std::unordered_map, long, bool >, 73
- cpp_std_map_like_to_py_dict< std::unordered_map, long, double >, 74
- cpp_std_map_like_to_py_dict< std::unordered_map, long, long >, 74
- cpp_std_map_like_to_py_dict< std::unordered_map, long, std::complex< double > >, 74
- cpp_std_map_like_to_py_dict< std::unordered_map, long, std::string >, 75
- cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u16string >, 75
- cpp_std_map_like_to_py_dict< std::unordered_map, long, std::u32string >, 76
- cpp_std_map_like_to_py_dict< std::unordered_map, long, std::vector< char > >, 76
- cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, bool >, 76
- cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, double >, 77
- cpp_std_map_like_to_py_dict< std::unordered_map, std::complex< double >, long >, 77
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, bool >, 78
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, double >, 78
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, long >, 78
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::complex< double > >, 79
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::string >, 79
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::u16string >, 80
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::u32string >, 80
- cpp_std_map_like_to_py_dict< std::unordered_map, std::string, std::vector< char > >, 80
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, bool >, 81
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, double >, 81
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, long >, 82
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::complex< double > >, 82
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::string >, 82
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::u16string >, 83
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::u32string >, 83
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u16string, std::vector< char > >, 84
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, bool >, 84
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, double >, 84
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, long >, 85
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::complex< double > >, 85
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::string >, 86
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u16string >, 86
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::u32string >, 86
- cpp_std_map_like_to_py_dict< std::unordered_map, std::u32string, std::vector< char > >, 87
- cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, bool >, 87
- cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, double >, 88
- cpp_std_map_like_to_py_dict< std::unordered_map, std::vector< char >, long >, 88
- cpp_std_unordered_set_to_py_frozenset, 88
- cpp_std_unordered_set_to_py_frozenset< bool >, 89
- cpp_std_unordered_set_to_py_frozenset< double >, 89
- cpp_std_unordered_set_to_py_frozenset< long >, 90
- cpp_std_unordered_set_to_py_frozenset< std::complex< double > >, 90
- cpp_std_unordered_set_to_py_frozenset< std::string >, 90
- cpp_std_unordered_set_to_py_frozenset< std::u16string >, 91
- cpp_std_unordered_set_to_py_frozenset< std::u32string >, 91
- cpp_std_unordered_set_to_py_frozenset< std::vector< char > >, 91
- cpp_std_unordered_set_to_py_set, 92
- cpp_std_unordered_set_to_py_set< bool >, 92
- cpp_std_unordered_set_to_py_set< double >, 93
- cpp_std_unordered_set_to_py_set< long >, 93
- cpp_std_unordered_set_to_py_set< std::complex< double > >, 93
- cpp_std_unordered_set_to_py_set< std::string >, 94
- cpp_std_unordered_set_to_py_set< std::u16string >, 94
- cpp_std_unordered_set_to_py_set< std::u32string >, 95
- cpp_std_unordered_set_to_py_set< std::vector< char > >, 95

- cpp_string_to_py_bytearray, [95](#)
- cpp_string_to_py_unicode8, [95](#)
- cpp_u16string_to_py_unicode16, [96](#)
- cpp_u32string_to_py_unicode32, [97](#)
- cpp_vector_char_to_py_bytearray, [97](#)
- cpp_vector_char_to_py_bytes, [98](#)
- ErrorReturnValue, [29](#)
- FAIL_CONTAINER_KEY_WRONG_TYPE, [29](#)
- FAIL_CONTAINER_MEMBER_WRONG_TYPE, [29](#)
- FAIL_CONTAINER_VALUE_WRONG_TYPE, [29](#)
- FAIL_CONTAINER_WRONG_TYPE, [29](#)
- generic_cpp_std_list_like_to_py_list, [98](#)
- generic_cpp_std_list_like_to_py_list_like, [99](#)
- generic_cpp_std_list_like_to_py_tuple, [99](#), [100](#)
- generic_cpp_std_list_to_py_list, [100](#)
- generic_cpp_std_list_to_py_tuple, [101](#)
- generic_cpp_std_map_like_to_py_dict, [101](#)
- generic_cpp_std_unordered_set_to_py_frozenset, [102](#)
- generic_cpp_std_unordered_set_to_py_set, [102](#)
- generic_cpp_std_unordered_set_to_py_set_or_frozenset, [103](#)
- generic_cpp_std_vector_to_py_list, [103](#)
- generic_cpp_std_vector_to_py_tuple, [103](#)
- generic_py_dict_to_cpp_std_map_like, [103](#)
- generic_py_frozenset_to_cpp_std_unordered_set, [104](#)
- generic_py_list_to_cpp_std_list, [105](#)
- generic_py_list_to_cpp_std_list_like, [105](#)
- generic_py_list_to_cpp_std_vector, [106](#)
- generic_py_set_or_frozenset_to_cpp_std_unordered_set, [106](#)
- generic_py_set_to_cpp_std_unordered_set, [107](#)
- generic_py_tuple_to_cpp_std_list, [107](#)
- generic_py_tuple_to_cpp_std_list_like, [107](#), [108](#)
- generic_py_tuple_to_cpp_std_vector, [109](#)
- generic_py_unary_to_cpp_std_list_like, [109](#)
- py_bool_check, [109](#)
- py_bool_to_cpp_bool, [109](#)
- py_bytearray_check, [110](#)
- py_bytearray_to_cpp_string, [110](#)
- py_bytearray_to_cpp_vector_char, [110](#)
- py_bytes_check, [111](#)
- py_bytes_to_cpp_vector_char, [111](#)
- py_complex_check, [111](#)
- py_complex_to_cpp_complex, [112](#)
- py_dict_to_cpp_std_map_like, [112](#), [113](#)
- py_dict_to_cpp_std_map_like< std::map, bool, bool >, [113](#)
- py_dict_to_cpp_std_map_like< std::map, bool, double >, [113](#)
- py_dict_to_cpp_std_map_like< std::map, bool, long >, [114](#)
- py_dict_to_cpp_std_map_like< std::map, bool, std::complex< double > >, [114](#)
- py_dict_to_cpp_std_map_like< std::map, bool, std::string >, [115](#)
- py_dict_to_cpp_std_map_like< std::map, bool, std::u16string >, [115](#)
- py_dict_to_cpp_std_map_like< std::map, bool, std::u32string >, [116](#)
- py_dict_to_cpp_std_map_like< std::map, bool, std::vector< char > >, [116](#)
- py_dict_to_cpp_std_map_like< std::map, double, bool >, [116](#)
- py_dict_to_cpp_std_map_like< std::map, double, double >, [117](#)
- py_dict_to_cpp_std_map_like< std::map, double, long >, [117](#)
- py_dict_to_cpp_std_map_like< std::map, double, std::complex< double > >, [118](#)
- py_dict_to_cpp_std_map_like< std::map, double, std::string >, [118](#)
- py_dict_to_cpp_std_map_like< std::map, double, std::u16string >, [119](#)
- py_dict_to_cpp_std_map_like< std::map, double, std::u32string >, [119](#)
- py_dict_to_cpp_std_map_like< std::map, double, std::vector< char > >, [119](#)
- py_dict_to_cpp_std_map_like< std::map, long, bool >, [120](#)
- py_dict_to_cpp_std_map_like< std::map, long, CppCustomObject >, [120](#)
- py_dict_to_cpp_std_map_like< std::map, long, double >, [120](#)
- py_dict_to_cpp_std_map_like< std::map, long, long >, [121](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::complex< double > >, [121](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::string >, [122](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::u16string >, [122](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::u32string >, [123](#)
- py_dict_to_cpp_std_map_like< std::map, long, std::vector< char > >, [123](#)
- py_dict_to_cpp_std_map_like< std::map, std::complex< double >, bool >, [123](#)
- py_dict_to_cpp_std_map_like< std::map, std::complex< double >, double >, [124](#)
- py_dict_to_cpp_std_map_like< std::map, std::complex< double >, long >, [124](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, bool >, [125](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, double >, [125](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, long >, [126](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::complex< double > >, [126](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::string >, [127](#)
- py_dict_to_cpp_std_map_like< std::map, std::string, std::u16string >, [127](#)

[illegible]

- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, long >, 154
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::complex< double > >, 155
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::string >, 155
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u16string >, 155
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::u32string >, 156
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u16string, std::vector< char > >, 156
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, bool >, 157
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, double >, 157
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, long >, 158
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::complex< double > >, 158
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::string >, 159
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u16string >, 159
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::u32string >, 159
- py_dict_to_cpp_std_map_like< std::unordered_map, std::u32string, std::vector< char > >, 160
- py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, bool >, 160
- py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, double >, 161
- py_dict_to_cpp_std_map_like< std::unordered_map, std::vector< char >, long >, 161
- py_float_check, 162
- py_float_to_cpp_double, 162
- py_frozenset_check, 162
- py_frozenset_to_cpp_std_unordered_set, 163
- py_frozenset_to_cpp_std_unordered_set< bool >, 163
- py_frozenset_to_cpp_std_unordered_set< double >, 164
- py_frozenset_to_cpp_std_unordered_set< long >, 164
- py_frozenset_to_cpp_std_unordered_set< std::complex< double > >, 165
- py_frozenset_to_cpp_std_unordered_set< std::string >, 165
- py_frozenset_to_cpp_std_unordered_set< std::u16string >, 165
- py_frozenset_to_cpp_std_unordered_set< std::u32string >, 166
- py_frozenset_to_cpp_std_unordered_set< std::vector< char > >, 166
- py_list_check, 167
- py_list_get, 167
- py_list_len, 167
- py_list_new, 168
- py_list_set, 168
- py_list_to_cpp_std_list_like, 169
- py_list_to_cpp_std_list_like< bool >, 170
- py_list_to_cpp_std_list_like< CppCustomObject >, 170
- py_list_to_cpp_std_list_like< double >, 171
- py_list_to_cpp_std_list_like< long >, 171, 172
- py_list_to_cpp_std_list_like< std::complex< double > >, 172, 173
- py_list_to_cpp_std_list_like< std::string >, 173
- py_list_to_cpp_std_list_like< std::u16string >, 174
- py_list_to_cpp_std_list_like< std::u32string >, 175
- py_list_to_cpp_std_list_like< std::vector< char > >, 175, 176
- py_long_check, 176
- py_long_to_cpp_long, 177
- py_set_check, 177
- py_set_to_cpp_std_unordered_set, 177
- py_set_to_cpp_std_unordered_set< bool >, 178
- py_set_to_cpp_std_unordered_set< double >, 178
- py_set_to_cpp_std_unordered_set< long >, 179
- py_set_to_cpp_std_unordered_set< std::complex< double > >, 179
- py_set_to_cpp_std_unordered_set< std::string >, 179
- py_set_to_cpp_std_unordered_set< std::u16string >, 180
- py_set_to_cpp_std_unordered_set< std::u32string >, 180
- py_set_to_cpp_std_unordered_set< std::vector< char > >, 181
- py_tuple_check, 181
- py_tuple_get, 181
- py_tuple_len, 182
- py_tuple_new, 182
- py_tuple_set, 183
- py_tuple_to_cpp_std_list_like, 183, 184
- py_tuple_to_cpp_std_list_like< bool >, 184, 185
- py_tuple_to_cpp_std_list_like< double >, 185, 186
- py_tuple_to_cpp_std_list_like< long >, 186
- py_tuple_to_cpp_std_list_like< std::complex< double > >, 187
- py_tuple_to_cpp_std_list_like< std::string >, 188
- py_tuple_to_cpp_std_list_like< std::u16string >, 188, 189
- py_tuple_to_cpp_std_list_like< std::u32string >, 189, 190
- py_tuple_to_cpp_std_list_like< std::vector< char > >, 190
- py_unicode16_check, 191
- py_unicode16_to_cpp_u16string, 191
- py_unicode32_check, 192
- py_unicode32_to_cpp_u32string, 192
- py_unicode8_check, 192
- py_unicode8_to_cpp_string, 193

- SUCCESS, 29
- very_generic_cpp_std_list_like_to_py_unary, 193
- very_generic_py_unary_to_cpp_std_list_like, 194
- PYTHON_CPP_CONTAINERS_VERSION
 - python_convert.h, 271
- REGEX_SPACE_ANYTHING
 - TestFramework.cpp, 236
- REGEX_SPACE_FLOAT
 - TestFramework.cpp, 237
- REGEX_SPACE_INTEGER
 - TestFramework.cpp, 237
- REGEX_SPACE_STRING_NO_SPACE
 - TestFramework.cpp, 237
- REPORT_OK_OR_FAIL
 - test_common.h, 304
- REPORT_TEST_OUTPUT
 - test_common.h, 304
- REPORT_TEST_OUTPUT_WITH_CONTAINER_TYPE_STRING_LENGTH
 - test_common.h, 305
- REPORT_TEST_OUTPUT_WITH_STRING_LENGTH
 - test_common.h, 305
- REPORT_TEST_OUTPUT_WITH_TYPE
 - test_common.h, 305
- REQUIRED_INCLUDES
 - src.py.code_gen, 202
- reset_count_of_unique_string
 - TestFramework.cpp, 239
 - TestFramework.h, 243
- results
 - TestResultS, 228
- reverse_dict_names
 - cUserDefined.cpp, 411
- reverse_list_names
 - cUserDefined.cpp, 411
- reverse_vector
 - cPyCppContainers.cpp, 406
- rss_initial_mb
 - RSSSnapshot, 215
- rss_initial_pages
 - RSSSnapshot, 216
- rss_now_diff_mb
 - RSSSnapshot, 216
- rss_now_diff_pages
 - RSSSnapshot, 216
- rss_now_mb
 - RSSSnapshot, 216
- rss_now_pages
 - RSSSnapshot, 216
- rss_peak_diff_mb
 - RSSSnapshot, 216
- rss_peak_diff_pages
 - RSSSnapshot, 216
- rss_peak_initial_mb
 - RSSSnapshot, 216
- rss_peak_initial_pages
 - RSSSnapshot, 217
- rss_peak_now_mb
 - RSSSnapshot, 217
- rss_peak_now_pages
 - RSSSnapshot, 217
- RSS_SNAPSHOT
 - test_common.h, 305
- RSS_SNAPSHOT_REPORT
 - test_common.h, 305
- RSS_SNAPSHOT_REPORT_PAGES
 - get_rss.cpp, 232
- RSS_SNAPSHOT_WITH_CONTAINER_TYPE_AND_TYPE
 - test_common.h, 306
- RSS_SNAPSHOT_WITH_TYPE
 - test_common.h, 306
- RSS_SNAPSHOT_WITHOUT_TYPE
 - test_common.h, 306
- RSSSnapshot, 214
 - m_name, 217
 - m_rss_initial, 217
 - m_rss_peak_initial, 217
 - m_name, 215
 - rss_initial_mb, 215
 - rss_initial_pages, 216
 - rss_now_diff_mb, 216
 - rss_now_diff_pages, 216
 - rss_now_mb, 216
 - rss_now_pages, 216
 - rss_peak_diff_mb, 216
 - rss_peak_diff_pages, 216
 - rss_peak_initial_mb, 216
 - rss_peak_initial_pages, 217
 - rss_peak_now_mb, 217
 - rss_peak_now_pages, 217
- scaleValues
 - TestResult, 226
- seconds
 - ExecClock, 212
- SET_RESULT_IF_PY_ERR_OCCURRED
 - test_common.h, 306
- setFailed
 - TestResult, 226
- SINGLE_ARGUMENT_METHOD
 - cPyCppContainers.cpp, 391
- src, 195
- src.py, 195
- src.py.code_gen, 195
 - AUTO_FILE_NAME, 198
 - CPP_MAP_TYPE_TO_PY_DICT_BASE_DECL, 198
 - CPP_MAP_TYPE_TO_PY_DICT_DECL, 198
 - CPP_MAP_TYPE_TO_PY_DICT_DEFN, 199
 - CPP_MAP_TYPES, 199
 - CPP_NAMESPACE, 199
 - CPP_PY_DICT_TO_MAP_TYPE_BASE_DECL, 199
 - CPP_PY_DICT_TO_MAP_TYPE_DECL, 199
 - CPP_PY_DICT_TO_MAP_TYPE_DEFN, 200
 - CPP_TYPE_TO_FUNCS, 200

- CPP_TYPES_TO_EXCLUDE_BY_CPP_CONTAINERS, 200
- CPP_UNARY_FUNCTION_TO_PY_BASE_DECL, 201
- CPP_UNARY_FUNCTION_TO_PY_DECL, 201
- CPP_UNARY_FUNCTION_TO_PY_DEFN, 201
- declarations, 196
- definitions, 197
- defn_name_from_decl_name, 197
- dict_map_declarations, 197
- dict_map_definitions, 197
- logger, 201
- main, 197
- PROJECT_VERSION, 201
- PY_TO_CPP_UNARY_FUNCTION_BASE_DECL, 202
- PY_TO_CPP_UNARY_FUNCTION_DECL, 202
- PY_TO_CPP_UNARY_FUNCTION_DEFN, 202
- REQUIRED_INCLUDES, 202
- UNARY_COLLECTIONS, 202
- unary_declarations, 198
- unary_definitions, 198
- write_files, 198
- src.py.code_gen.CodeCount, 209
- src.py.code_gen_common, 203
- src.py.code_gen_common.CppTypeFunctions, 211
- src.py.code_gen_common.TypeConversionFunctions, 229
- src.py.code_gen_common.UnaryFunctions, 229
- src.py.code_gen_documentation, 203
 - comment_list_str, 204
 - comment_str, 204
 - cpp_comment_section, 204
 - documentation, 204
 - doxygen_cpp_to_python_dict_base_class, 204
 - doxygen_cpp_to_python_dict_instantiation, 205
 - doxygen_cpp_to_python_unary_base_class, 205
 - doxygen_cpp_to_python_unary_instantiation, 205
 - doxygen_python_dict_to_cpp_base_class, 206
 - doxygen_python_dict_to_cpp_instantiation, 206
 - doxygen_python_to_cpp_unary_base_class, 207
 - doxygen_python_to_cpp_unary_instantiation, 207
 - get_codegen_please_no_edit_warning, 207
 - get_codegen_please_no_edit_warning_context, 208
 - WIDTH, 208
- std, 208
- std::hash< std::complex< double > >, 213
 - operator(), 213
- std::hash< std::vector< char > >, 213
 - operator(), 214
- std::less< std::complex< T > >, 214
 - operator(), 214
- str_count
 - TestFramework.cpp, 240
- StreamFormatState, 218
 - ~StreamFormatState, 218
 - StreamFormatState, 218
- SubTestCount, 219
 - failure, 219
 - m_failure, 220
 - m_test_count, 220
 - SubTestCount, 219
 - test, 219
 - test_count, 220
 - test_failures, 220
- SUCCESS
 - Python_Cpp_Containers, 29
- test
 - SubTestCount, 219
- test_all
 - main.cpp, 415
- test_bool_to_py_bool_multiple
 - test_performance.cpp, 359
- test_common.cpp
 - compare_dict< std::map, std::string, std::string >, 277
 - compare_dict< std::map, std::u16string, std::u16string >, 277
 - compare_dict< std::map, std::u32string, std::u32string >, 278
 - compare_dict< std::unordered_map, std::string, std::string >, 278
 - compare_dict< std::unordered_map, std::u16string, std::u16string >, 278
 - compare_dict< std::unordered_map, std::u32string, std::u32string >, 278
 - compare_list< bool >, 278
 - compare_list< double >, 278
 - compare_list< long >, 279
 - compare_list< std::complex< double > >, 279
 - compare_list< std::string >, 279
 - compare_list< std::u16string >, 279
 - compare_list< std::u32string >, 279
 - compare_list< std::vector< char > >, 279
 - compare_set< std::string >, 280
 - compare_set< std::u16string >, 280
 - compare_set< std::u32string >, 280
 - compare_set< std::vector< char > >, 280
 - compare_tuple< bool >, 280
 - compare_tuple< double >, 280
 - compare_tuple< long >, 281
 - compare_tuple< std::complex< double > >, 281
 - compare_tuple< std::string >, 281
 - compare_tuple< std::u16string >, 281
 - compare_tuple< std::u32string >, 281
 - compare_tuple< std::vector< char > >, 281
 - new_py_dict_bytes, 282
 - new_py_dict_string, 282
 - new_py_dict_string16, 282
 - new_py_dict_string32, 283
 - new_py_list_bytes, 283
 - new_py_list_string, 284
 - new_py_list_string16, 284
 - new_py_list_string32, 284
 - new_py_set_bytes, 285

- new_py_set_string, 285
- new_py_set_u16string, 285
- new_py_set_u32string, 286
- new_py_tuple_bytes, 286
- new_py_tuple_string, 287
- new_py_tuple_string16, 287
- new_py_tuple_string32, 287
- test_cpp_std_map_like_to_py_dict_bytes, 288
- test_cpp_std_map_like_to_py_dict_string, 288
- test_cpp_std_map_like_to_py_dict_string16, 288
- test_cpp_std_map_like_to_py_dict_string32, 288
- test_cpp_std_map_to_py_dict_bytes, 288
- test_cpp_std_map_to_py_dict_string, 289
- test_cpp_std_map_to_py_dict_string16, 289
- test_cpp_std_map_to_py_dict_string32, 289
- test_cpp_std_unordered_map_to_py_dict_bytes, 289
- test_cpp_std_unordered_map_to_py_dict_string, 289
- test_cpp_std_unordered_map_to_py_dict_string16, 289
- test_cpp_std_unordered_map_to_py_dict_string32, 290
- test_py_dict_to_cpp_std_map_bytes, 290
- test_py_dict_to_cpp_std_map_like_bytes, 290
- test_py_dict_to_cpp_std_map_like_string, 290
- test_py_dict_to_cpp_std_map_like_string16, 290
- test_py_dict_to_cpp_std_map_like_string32, 291
- test_py_dict_to_cpp_std_map_string, 291
- test_py_dict_to_cpp_std_map_string16, 291
- test_py_dict_to_cpp_std_map_string32, 291
- test_py_dict_to_cpp_std_unordered_map_bytes, 291
- test_py_dict_to_cpp_std_unordered_map_string, 292
- test_py_dict_to_cpp_std_unordered_map_u16string, 292
- test_py_dict_to_cpp_std_unordered_map_u32string, 292
- test_py_list_bytes_to_vector, 292
- test_py_list_str16_to_vector, 293
- test_py_list_str32_to_vector, 293
- test_py_list_str_to_vector, 293
- test_py_set_bytes_to_unordered_set, 294
- test_py_set_string16_to_unordered_set, 294
- test_py_set_string32_to_unordered_set, 294
- test_py_set_string_to_unordered_set, 294
- test_py_tuple_bytes_to_vector, 295
- test_py_tuple_str16_to_vector, 295
- test_py_tuple_str32_to_vector, 295
- test_py_tuple_str_to_vector, 296
- test_unordered_set_bytes_to_py_set, 296
- test_unordered_set_string_to_py_set, 296
- test_unordered_set_u16string_to_py_set, 297
- test_unordered_set_u32string_to_py_set, 297
- test_vector_string_to_py_list, 297
- test_vector_string_to_py_tuple, 297
- test_vector_u16string_to_py_list, 298
- test_vector_u16string_to_py_tuple, 298
- test_vector_u32string_to_py_list, 299
- test_vector_u32string_to_py_tuple, 299
- test_vector_vector_char_to_py_list, 300
- test_vector_vector_char_to_py_tuple, 300
- test_common.h
 - compare_dict, 307, 308
 - compare_dict< std::map, std::string, std::string >, 308
 - compare_dict< std::map, std::u16string, std::u16string >, 308
 - compare_dict< std::map, std::u32string, std::u32string >, 309
 - compare_dict< std::unordered_map, std::string, std::string >, 309
 - compare_dict< std::unordered_map, std::u16string, std::u16string >, 309
 - compare_dict< std::unordered_map, std::u32string, std::u32string >, 309
 - compare_list, 309, 310
 - compare_list< bool >, 310
 - compare_list< double >, 311
 - compare_list< long >, 311
 - compare_list< std::complex< double > >, 311
 - compare_list< std::string >, 311
 - compare_list< std::vector< char > >, 311
 - compare_set, 311, 312
 - compare_set< std::string >, 313
 - compare_set< std::u16string >, 313
 - compare_set< std::u32string >, 313
 - compare_set< std::vector< char > >, 313
 - compare_tuple, 313, 314
 - compare_tuple< bool >, 314
 - compare_tuple< double >, 315
 - compare_tuple< long >, 315
 - compare_tuple< std::complex< double > >, 315
 - compare_tuple< std::string >, 315
 - compare_tuple< std::u16string >, 315
 - compare_tuple< std::u32string >, 315
 - compare_tuple< std::vector< char > >, 316
 - compare_tuple_or_list, 316
 - new_py_dict_bytes, 316
 - new_py_dict_string, 317
 - new_py_dict_string16, 317
 - new_py_dict_string32, 318
 - new_py_list_bytes, 318
 - new_py_list_string, 318
 - new_py_list_string16, 319
 - new_py_list_string32, 319
 - new_py_set_bytes, 319
 - new_py_set_string, 320
 - new_py_set_u16string, 320
 - new_py_set_u32string, 321
 - new_py_tuple_bytes, 321
 - new_py_tuple_string, 321
 - new_py_tuple_string16, 322
 - new_py_tuple_string32, 322
 - PY_ERR_ON_ENTRY_RETURN_CODE, 339

- PY_ERR_ON_EXIT_RETURN_CODE, 339
- REPORT_OK_OR_FAIL, 304
- REPORT_TEST_OUTPUT, 304
- REPORT_TEST_OUTPUT_WITH_CONTAINER_TYPE_STRING_LENGTH, 305
- REPORT_TEST_OUTPUT_WITH_STRING_LENGTH, 305
- REPORT_TEST_OUTPUT_WITH_TYPE, 305
- RSS_SNAPSHOT, 305
- RSS_SNAPSHOT_REPORT, 305
- RSS_SNAPSHOT_WITH_CONTAINER_TYPE_AND_TYPE_STRING_LENGTH, 306
- RSS_SNAPSHOT_WITH_TYPE, 306
- RSS_SNAPSHOT_WITHOUT_TYPE, 306
- SET_RESULT_IF_PY_ERR_OCCURRED, 306
- test_cpp_std_map_like_to_py_dict, 322
- test_cpp_std_map_to_py_dict, 323
- test_cpp_std_map_to_py_dict_bytes, 323
- test_cpp_std_map_to_py_dict_string, 323
- test_cpp_std_map_to_py_dict_string16, 323
- test_cpp_std_map_to_py_dict_string32, 323
- test_cpp_std_unordered_map_to_py_dict, 324
- test_cpp_std_unordered_map_to_py_dict_bytes, 324
- test_cpp_std_unordered_map_to_py_dict_string, 324
- test_cpp_std_unordered_map_to_py_dict_string16, 324
- test_cpp_std_unordered_map_to_py_dict_string32, 324
- TEST_FOR_PY_ERR_ON_ENTRY, 306
- TEST_FOR_PY_ERR_ON_EXIT, 307
- test_py_dict_to_cpp_std_map, 325
- test_py_dict_to_cpp_std_map_bytes, 325
- test_py_dict_to_cpp_std_map_like, 325
- test_py_dict_to_cpp_std_map_string, 325
- test_py_dict_to_cpp_std_map_string16, 325
- test_py_dict_to_cpp_std_map_string32, 326
- test_py_dict_to_cpp_std_unordered_map, 326
- test_py_dict_to_cpp_std_unordered_map_bytes, 326
- test_py_dict_to_cpp_std_unordered_map_string, 326
- test_py_dict_to_cpp_std_unordered_map_u16string, 326
- test_py_dict_to_cpp_std_unordered_map_u32string, 327
- test_py_list_bytes_to_vector, 327
- test_py_list_str16_to_vector, 327
- test_py_list_str32_to_vector, 328
- test_py_list_str_to_vector, 328
- test_py_list_to_vector, 329
- test_py_list_to_vector_round_trip, 329
- test_py_set_bytes_to_unordered_set, 329
- test_py_set_string16_to_unordered_set, 329
- test_py_set_string32_to_unordered_set, 329
- test_py_set_string_to_unordered_set, 329
- test_py_set_to_unordered_set, 330
- test_py_tuple_bytes_to_vector, 330
- test_py_tuple_str16_to_vector, 330
- test_py_tuple_str32_to_vector, 331
- test_py_tuple_to_vector, 332
- test_py_tuple_to_vector_round_trip, 332
- test_unordered_set_bytes_to_py_set, 333
- test_unordered_set_string_to_py_set, 333
- test_unordered_set_to_py_set, 333
- test_unordered_set_u16string_to_py_set, 333
- test_unordered_set_u32string_to_py_set, 333
- test_vector_string_to_py_list, 334
- test_vector_string_to_py_tuple, 334
- test_vector_to_py_list, 334
- test_vector_to_py_list_round_trip, 335
- test_vector_to_py_tuple, 335
- test_vector_to_py_tuple_round_trip, 336
- test_vector_u16string_to_py_list, 336
- test_vector_u16string_to_py_tuple, 337
- test_vector_u32string_to_py_list, 337
- test_vector_u32string_to_py_tuple, 338
- test_vector_vector_char_to_py_list, 338
- test_vector_vector_char_to_py_tuple, 338
- test_complex_to_py_complex_multiple
- test_performance.cpp, 359
- test_count
 - SubTestCount, 220
- test_cpp_std_map_like_to_py_dict
 - test_common.h, 322
- test_cpp_std_map_like_to_py_dict_bytes
 - test_common.cpp, 288
- test_cpp_std_map_like_to_py_dict_multiple
 - test_performance.cpp, 360
- test_cpp_std_map_like_to_py_dict_string
 - test_common.cpp, 288
- test_cpp_std_map_like_to_py_dict_string16
 - test_common.cpp, 288
- test_cpp_std_map_like_to_py_dict_string32
 - test_common.cpp, 288
- test_cpp_std_map_like_to_py_dict_string_multiple
 - test_performance.cpp, 360
- test_cpp_std_map_like_to_py_dict_vector_char_multiple
 - test_performance.cpp, 360
- test_cpp_std_map_to_py_dict
 - test_common.h, 323
- test_cpp_std_map_to_py_dict_bytes
 - test_common.cpp, 288
 - test_common.h, 323
- test_cpp_std_map_to_py_dict_multiple
 - test_performance.cpp, 360
- test_cpp_std_map_to_py_dict_string
 - test_common.cpp, 289
 - test_common.h, 323
- test_cpp_std_map_to_py_dict_string16
 - test_common.cpp, 289
 - test_common.h, 323
- test_cpp_std_map_to_py_dict_string32
 - test_common.cpp, 289

- test_common.h, 323
 - test_cpp_std_map_to_py_dict_string_multiple
 - test_performance.cpp, 360
 - test_cpp_std_map_to_py_dict_vector_char_multiple
 - test_performance.cpp, 361
 - test_cpp_std_unordered_map_to_py_dict
 - test_common.h, 324
 - test_cpp_std_unordered_map_to_py_dict_bytes
 - test_common.cpp, 289
 - test_common.h, 324
 - test_cpp_std_unordered_map_to_py_dict_multiple
 - test_performance.cpp, 361
 - test_cpp_std_unordered_map_to_py_dict_string
 - test_common.cpp, 289
 - test_common.h, 324
 - test_cpp_std_unordered_map_to_py_dict_string16
 - test_common.cpp, 289
 - test_common.h, 324
 - test_cpp_std_unordered_map_to_py_dict_string32
 - test_common.cpp, 290
 - test_common.h, 324
 - test_cpp_std_unordered_map_to_py_dict_string_multiple
 - test_performance.cpp, 361
 - test_cpp_std_unordered_map_to_py_dict_vector_char_multiple
 - test_performance.cpp, 361
 - test_cpp_string_to_py_str_multiple
 - test_performance.cpp, 361
 - test_cpp_u16string_to_py_str16_multiple
 - test_performance.cpp, 362
 - test_cpp_u32string_to_py_str32_multiple
 - test_performance.cpp, 362
 - test_cpp_vector_char_to_py_bytes_multiple
 - test_performance.cpp, 362
 - test_double_to_py_float_multiple
 - test_performance.cpp, 363
 - test_example_cpp_std_map_to_py_dict
 - test_functional.cpp, 340
 - test_example_cpp_std_unordered_map_to_py_dict
 - test_functional.cpp, 340
 - test_example_py_dict_to_cpp_std_unordered_map
 - test_functional.cpp, 340
 - test_example_py_tuple_to_vector_double
 - test_functional.cpp, 340
 - test_example_vector_to_py_tuple_double
 - test_functional.cpp, 340
 - test_failures
 - SubTestCount, 220
 - TEST_FOR_PY_ERR_ON_ENTRY
 - test_common.h, 306
 - TEST_FOR_PY_ERR_ON_EXIT
 - test_common.h, 307
 - test_functional.cpp
 - test_example_cpp_std_map_to_py_dict, 340
 - test_example_cpp_std_unordered_map_to_py_dict, 340
 - test_example_py_dict_to_cpp_std_unordered_map, 340
 - test_example_py_tuple_to_vector_double, 340
 - test_example_vector_to_py_tuple_double, 340
 - test_functional_all, 341
 - test_functional_dict_copy, 341
 - test_functional_dict_setitem, 341
 - test_functional_dict_with_std_map, 341
 - test_functional_dict_with_std_unordred_map, 341
 - test_functional_frozenset_add, 342
 - test_functional_frozenset_add_from_iterable, 342
 - test_functional_list, 342
 - test_functional_list_setitem, 342
 - test_functional_set, 343
 - test_functional_set_add, 343
 - test_functional_set_add_from_iterable, 343
 - test_functional_tuple, 343
 - test_functional_tuple_setitem, 343
 - test_functional.h
 - test_functional_all, 344
 - TEST_FUNCTIONAL_ALL
 - main.cpp, 414
 - test_functional_all
 - test_functional.cpp, 341
 - test_functional.h, 344
 - test_functional_dict_copy
 - test_functional.cpp, 341
 - test_functional_dict_setitem
 - test_functional.cpp, 341
 - test_functional_dict_with_std_map
 - test_functional.cpp, 341
 - test_functional_dict_with_std_unordred_map
 - test_functional.cpp, 341
 - test_functional_frozenset_add
 - test_functional.cpp, 342
 - test_functional_frozenset_add_from_iterable
 - test_functional.cpp, 342
 - test_functional_list
 - test_functional.cpp, 342
 - test_functional_list_setitem
 - test_functional.cpp, 342
 - test_functional_set
 - test_functional.cpp, 343
 - test_functional_set_add
 - test_functional.cpp, 343
 - test_functional_set_add_from_iterable
 - test_functional.cpp, 343
 - test_functional_tuple
 - test_functional.cpp, 343
 - test_functional_tuple_setitem
 - test_functional.cpp, 343
- test_internal.cpp
 - doubles_cmp, 345
 - test_internal_all, 345
 - test_internal_test_result_atomic_test_mean_exec_time, 345
 - test_internal_test_result_exec_time, 345
 - test_internal_test_result_exec_time_min_max, 346
 - test_internal_test_result_string, 346
 - test_internal_test_result_string_multiple_a, 346
 - test_internal_test_result_string_multiple_b, 346

- test_internal_test_result_string_using_rate, 346
- test_internal_test_result_test_count, 346
- test_internal_test_result_total_time, 347
- test_internal.h
 - test_internal_all, 347
- TEST_INTERNAL_ALL
 - main.cpp, 415
- test_internal_all
 - test_internal.cpp, 345
 - test_internal.h, 347
- test_internal_test_result_atomic_test_mean_exec_time
 - test_internal.cpp, 345
- test_internal_test_result_exec_time
 - test_internal.cpp, 345
- test_internal_test_result_exec_time_min_max
 - test_internal.cpp, 346
- test_internal_test_result_string
 - test_internal.cpp, 346
- test_internal_test_result_string_multiple_a
 - test_internal.cpp, 346
- test_internal_test_result_string_multiple_b
 - test_internal.cpp, 346
- test_internal_test_result_string_using_rate
 - test_internal.cpp, 346
- test_internal_test_result_test_count
 - test_internal.cpp, 346
- test_internal_test_result_total_time
 - test_internal.cpp, 347
- test_list_like_string_to_py_list_multiple
 - test_performance.cpp, 363
- test_list_like_string_to_py_tuple_multiple
 - test_performance.cpp, 363
- test_list_like_to_py_list_multiple
 - test_performance.cpp, 363
- test_list_like_to_py_tuple_multiple
 - test_performance.cpp, 363
- test_list_like_u16string_to_py_list_multiple
 - test_performance.cpp, 364
- test_list_like_u32string_to_py_list_multiple
 - test_performance.cpp, 364
- test_list_like_vector_char_to_py_list_multiple
 - test_performance.cpp, 364
- test_list_like_vector_char_to_py_tuple_multiple
 - test_performance.cpp, 364
- test_list_string_to_py_list_multiple
 - test_performance.cpp, 365
- test_list_string_to_py_tuple_multiple
 - test_performance.cpp, 365
- test_list_to_py_list_multiple
 - test_performance.cpp, 365
- test_list_to_py_tuple_multiple
 - test_performance.cpp, 365
- test_list_u16string_to_py_list_multiple
 - test_performance.cpp, 365
- test_list_u32string_to_py_list_multiple
 - test_performance.cpp, 366
- test_list_vector_char_to_py_list_multiple
 - test_performance.cpp, 366
- test_list_vector_char_to_py_tuple_multiple
 - test_performance.cpp, 366
- test_long_to_py_int_multiple
 - test_performance.cpp, 366
- test_memory.cpp
 - test_memory_all, 348
 - test_memory_py_dict, 348
 - test_memory_py_tuple_float, 349
 - test_memory_py_tuple_str16_to_vector, 349
 - test_memory_py_tuple_str32_to_vector, 349
 - test_memory_py_tuple_unicode8_to_vector, 349
 - test_memory_py_tuple_vector_char_to_vector, 349
 - test_memory_test_vector_string_to_py_tuple, 350
 - test_memory_vector_u16string_to_py_tuple, 350
 - test_memory_vector_u32string_to_py_tuple, 350
 - test_memory_vector_vector_char_to_py_set, 350
 - test_memory_vector_vector_char_to_py_set_special, 350
 - test_memory_vector_vector_char_to_py_tuple, 351
- test_memory.h
 - test_memory_all, 351
- TEST_MEMORY_ALL
 - main.cpp, 415
- test_memory_all
 - test_memory.cpp, 348
 - test_memory.h, 351
- test_memory_py_dict
 - test_memory.cpp, 348
- test_memory_py_tuple_float
 - test_memory.cpp, 349
- test_memory_py_tuple_str16_to_vector
 - test_memory.cpp, 349
- test_memory_py_tuple_str32_to_vector
 - test_memory.cpp, 349
- test_memory_py_tuple_unicode8_to_vector
 - test_memory.cpp, 349
- test_memory_py_tuple_vector_char_to_vector
 - test_memory.cpp, 349
- test_memory_test_vector_string_to_py_tuple
 - test_memory.cpp, 350
- test_memory_vector_u16string_to_py_tuple
 - test_memory.cpp, 350
- test_memory_vector_u32string_to_py_tuple
 - test_memory.cpp, 350
- test_memory_vector_vector_char_to_py_set
 - test_memory.cpp, 350
- test_memory_vector_vector_char_to_py_set_special
 - test_memory.cpp, 350
- test_memory_vector_vector_char_to_py_tuple
 - test_memory.cpp, 351
- test_perf_cpp_std_map_to_py_dict_multiple
 - test_performance.cpp, 366
- test_perf_cpp_std_map_to_py_dict_string_multiple
 - test_performance.cpp, 367
- test_perf_cpp_std_map_to_py_dict_vector_char_multiple
 - test_performance.cpp, 367

- test_perf_cpp_std_unordered_map_to_py_dict_multiple
test_performance.cpp, [367](#)
- test_perf_cpp_std_unordered_map_to_py_dict_string_multiple
test_performance.cpp, [367](#)
- test_perf_cpp_std_unordered_map_to_py_dict_vector_char_multiple
test_performance.cpp, [367](#)
- test_perf_list_string_to_py_list_multiple
test_performance.cpp, [367](#)
- test_perf_list_string_to_py_tuple_multiple
test_performance.cpp, [368](#)
- test_perf_list_to_py_list_multiple
test_performance.cpp, [368](#)
- test_perf_list_to_py_tuple_multiple
test_performance.cpp, [368](#)
- test_perf_list_u16string_to_py_list_multiple
test_performance.cpp, [368](#)
- test_perf_list_u32string_to_py_list_multiple
test_performance.cpp, [368](#)
- test_perf_list_vector_char_to_py_list_multiple
test_performance.cpp, [368](#)
- test_perf_list_vector_char_to_py_tuple_multiple
test_performance.cpp, [369](#)
- test_perf_py_dict_to_cpp_std_map_multiple
test_performance.cpp, [369](#)
- test_perf_py_dict_to_cpp_std_map_string_multiple
test_performance.cpp, [369](#)
- test_perf_py_dict_to_cpp_std_map_vector_char_multiple
test_performance.cpp, [369](#)
- test_perf_py_dict_to_cpp_std_unordered_map_multiple
test_performance.cpp, [369](#)
- test_perf_py_dict_to_cpp_std_unordered_map_string_multiple
test_performance.cpp, [369](#)
- test_perf_py_dict_to_cpp_std_unordered_map_vector_char_multiple
test_performance.cpp, [370](#)
- test_perf_py_list_to_list_multiple
test_performance.cpp, [370](#)
- test_perf_py_list_to_list_string_multiple
test_performance.cpp, [370](#)
- test_perf_py_list_to_list_u16string_multiple
test_performance.cpp, [370](#)
- test_perf_py_list_to_list_u32string_multiple
test_performance.cpp, [370](#)
- test_perf_py_list_to_list_vector_char_multiple
test_performance.cpp, [370](#)
- test_perf_py_list_to_vector_multiple
test_performance.cpp, [371](#)
- test_perf_py_list_to_vector_string_multiple
test_performance.cpp, [371](#)
- test_perf_py_list_to_vector_u16string_multiple
test_performance.cpp, [371](#)
- test_perf_py_list_to_vector_u32string_multiple
test_performance.cpp, [371](#)
- test_perf_py_list_to_vector_vector_char_multiple
test_performance.cpp, [371](#)
- test_perf_py_set_bytes_to_unordered_set_vector_char_multiple
test_performance.cpp, [371](#)
- test_perf_py_set_str16_to_unordered_set_u16string_multiple
test_performance.cpp, [372](#)
- test_perf_py_set_str32_to_unordered_set_u32string_multiple
test_performance.cpp, [372](#)
- test_perf_py_set_str_to_unordered_set_string_multiple
test_performance.cpp, [372](#)
- test_perf_py_set_to_unordered_set_multiple
test_performance.cpp, [372](#)
- test_perf_py_tuple_to_list_multiple
test_performance.cpp, [372](#)
- test_perf_py_tuple_to_list_string_multiple
test_performance.cpp, [373](#)
- test_perf_py_tuple_to_list_vector_char_multiple
test_performance.cpp, [373](#)
- test_perf_py_tuple_to_vector_multiple
test_performance.cpp, [373](#)
- test_perf_py_tuple_to_vector_string_multiple
test_performance.cpp, [373](#)
- test_perf_py_tuple_to_vector_vector_char_multiple
test_performance.cpp, [373](#)
- test_perf_unordered_set_string_to_py_set_multiple
test_performance.cpp, [373](#)
- test_perf_unordered_set_to_py_set_multiple
test_performance.cpp, [374](#)
- test_perf_unordered_set_u16string_to_py_set_multiple
test_performance.cpp, [374](#)
- test_perf_unordered_set_u32string_to_py_set_multiple
test_performance.cpp, [374](#)
- test_perf_unordered_set_vector_char_to_py_set_multiple
test_performance.cpp, [374](#)
- test_perf_vector_string_to_py_list_multiple
test_performance.cpp, [374](#)
- test_perf_vector_string_to_py_tuple_multiple
test_performance.cpp, [374](#)
- test_perf_vector_to_py_list_multiple
test_performance.cpp, [375](#)
- test_perf_vector_to_py_tuple_multiple
test_performance.cpp, [375](#)
- test_perf_vector_u16string_to_py_list_multiple
test_performance.cpp, [375](#)
- test_perf_vector_u32string_to_py_list_multiple
test_performance.cpp, [375](#)
- test_perf_vector_vector_char_to_py_list_multiple
test_performance.cpp, [375](#)
- test_perf_vector_vector_char_to_py_tuple_multiple
test_performance.cpp, [375](#)
- test_performance.cpp
INC_SIZE_OF_CONTAINER_MULTIPLE, [388](#)
INC_STRING_LENGTH_MULTIPLE, [388](#)
LIMIT_SIZE_OF_CONTAINER, [388](#)
LIMIT_SIZE_OF_CONTAINER_DICT, [388](#)
LIMIT_STRING_LENGTH, [388](#)
MIN_SIZE_OF_CONTAINER, [388](#)
MIN_STRING_LENGTH_HASHABLE, [388](#)
MIN_STRING_LENGTH_NON_HASHABLE, [389](#)
test_bool_to_py_bool_multiple, [359](#)
test_complex_to_py_complex_multiple, [359](#)
test_cpp_std_map_like_to_py_dict_multiple, [360](#)
test_cpp_std_map_like_to_py_dict_string_multiple, [360](#)

- test_cpp_std_map_like_to_py_dict_vector_char_multiple, 360
- test_cpp_std_map_to_py_dict_multiple, 360
- test_cpp_std_map_to_py_dict_string_multiple, 360
- test_cpp_std_map_to_py_dict_vector_char_multiple, 361
- test_cpp_std_unordered_map_to_py_dict_multiple, 361
- test_cpp_std_unordered_map_to_py_dict_string_multiple, 361
- test_cpp_std_unordered_map_to_py_dict_vector_char_multiple, 361
- test_cpp_string_to_py_str_multiple, 361
- test_cpp_u16string_to_py_str16_multiple, 362
- test_cpp_u32string_to_py_str32_multiple, 362
- test_cpp_vector_char_to_py_bytes_multiple, 362
- test_double_to_py_float_multiple, 363
- test_list_like_string_to_py_list_multiple, 363
- test_list_like_string_to_py_tuple_multiple, 363
- test_list_like_to_py_list_multiple, 363
- test_list_like_to_py_tuple_multiple, 363
- test_list_like_u16string_to_py_list_multiple, 364
- test_list_like_u32string_to_py_list_multiple, 364
- test_list_like_vector_char_to_py_list_multiple, 364
- test_list_like_vector_char_to_py_tuple_multiple, 364
- test_list_string_to_py_list_multiple, 365
- test_list_string_to_py_tuple_multiple, 365
- test_list_to_py_list_multiple, 365
- test_list_to_py_tuple_multiple, 365
- test_list_u16string_to_py_list_multiple, 365
- test_list_u32string_to_py_list_multiple, 366
- test_list_vector_char_to_py_list_multiple, 366
- test_list_vector_char_to_py_tuple_multiple, 366
- test_long_to_py_int_multiple, 366
- test_perf_cpp_std_map_to_py_dict_multiple, 366
- test_perf_cpp_std_map_to_py_dict_string_multiple, 367
- test_perf_cpp_std_map_to_py_dict_vector_char_multiple, 367
- test_perf_cpp_std_unordered_map_to_py_dict_multiple, 367
- test_perf_cpp_std_unordered_map_to_py_dict_string_multiple, 367
- test_perf_cpp_std_unordered_map_to_py_dict_vector_char_multiple, 367
- test_perf_list_string_to_py_list_multiple, 367
- test_perf_list_string_to_py_tuple_multiple, 368
- test_perf_list_to_py_list_multiple, 368
- test_perf_list_to_py_tuple_multiple, 368
- test_perf_list_u16string_to_py_list_multiple, 368
- test_perf_list_u32string_to_py_list_multiple, 368
- test_perf_list_vector_char_to_py_list_multiple, 368
- test_perf_list_vector_char_to_py_tuple_multiple, 369
- test_perf_py_dict_to_cpp_std_map_multiple, 369
- test_perf_py_dict_to_cpp_std_map_string_multiple, 369
- test_perf_py_dict_to_cpp_std_map_vector_char_multiple, 369
- test_perf_py_dict_to_cpp_std_unordered_map_multiple, 369
- test_perf_py_dict_to_cpp_std_unordered_map_string_multiple, 369
- test_perf_py_dict_to_cpp_std_unordered_map_vector_char_multiple, 370
- test_perf_py_list_to_list_multiple, 370
- test_perf_py_list_to_list_string_multiple, 370
- test_perf_py_list_to_list_u16string_multiple, 370
- test_perf_py_list_to_list_u32string_multiple, 370
- test_perf_py_list_to_list_vector_char_multiple, 370
- test_perf_py_list_to_vector_multiple, 371
- test_perf_py_list_to_vector_string_multiple, 371
- test_perf_py_list_to_vector_u16string_multiple, 371
- test_perf_py_list_to_vector_u32string_multiple, 371
- test_perf_py_list_to_vector_vector_char_multiple, 371
- test_perf_py_set_bytes_to_unordered_set_vector_char_multiple, 371
- test_perf_py_set_str16_to_unordered_set_u16string_multiple, 372
- test_perf_py_set_str32_to_unordered_set_u32string_multiple, 372
- test_perf_py_set_str_to_unordered_set_string_multiple, 372
- test_perf_py_set_to_unordered_set_multiple, 372
- test_perf_py_tuple_to_list_multiple, 372
- test_perf_py_tuple_to_list_string_multiple, 373
- test_perf_py_tuple_to_list_vector_char_multiple, 373
- test_perf_py_tuple_to_vector_multiple, 373
- test_perf_py_tuple_to_vector_string_multiple, 373
- test_perf_py_tuple_to_vector_vector_char_multiple, 373
- test_perf_unordered_set_string_to_py_set_multiple, 373
- test_perf_unordered_set_to_py_set_multiple, 374
- test_perf_unordered_set_u16string_to_py_set_multiple, 374
- test_perf_unordered_set_u32string_to_py_set_multiple, 374
- test_perf_unordered_set_vector_char_to_py_set_multiple, 374
- test_perf_vector_string_to_py_list_multiple, 374
- test_perf_vector_string_to_py_tuple_multiple, 374
- test_perf_vector_to_py_list_multiple, 375
- test_perf_vector_to_py_tuple_multiple, 375
- test_perf_vector_u16string_to_py_list_multiple, 375
- test_perf_vector_u32string_to_py_list_multiple, 375
- test_perf_vector_vector_char_to_py_list_multiple, 375
- test_perf_vector_vector_char_to_py_tuple_multiple, 375

- 375
- test_performance_all, 376
- TEST_PERFORMANCE_DICTS, 357
- TEST_PERFORMANCE_FUNDAMENTAL_TYPES, 358
- TEST_PERFORMANCE_LISTS, 358
- TEST_PERFORMANCE_OBJECT_BOOL, 358
- TEST_PERFORMANCE_OBJECT_BYTES, 358
- TEST_PERFORMANCE_OBJECT_COMPLEX, 358
- TEST_PERFORMANCE_OBJECT_DOUBLE, 358
- TEST_PERFORMANCE_OBJECT_LONG, 358
- TEST_PERFORMANCE_OBJECT_STRING, 358
- TEST_PERFORMANCE_OBJECT_STRING_16, 359
- TEST_PERFORMANCE_OBJECT_STRING_32, 359
- TEST_PERFORMANCE_SETS, 359
- TEST_PERFORMANCE_TUPLES, 359
- test_py_bool_to_cpp_bool_multiple, 376
- test_py_bytes_to_cpp_vector_char_multiple, 376
- test_py_complex_to_cpp_complex_multiple, 376
- test_py_dict_to_cpp_std_map_like_multiple, 376
- test_py_dict_to_cpp_std_map_like_string_multiple, 376
- test_py_dict_to_cpp_std_map_like_vector_char_multiple, 377
- test_py_dict_to_cpp_std_map_multiple, 377
- test_py_dict_to_cpp_std_map_string_multiple, 377
- test_py_dict_to_cpp_std_map_vector_char_multiple, 377
- test_py_dict_to_cpp_std_unordered_map_multiple, 377
- test_py_dict_to_cpp_std_unordered_map_string_multiple, 378
- test_py_dict_to_cpp_std_unordered_map_vector_char_multiple, 378
- test_py_float_to_cpp_double_multiple, 378
- test_py_int_to_cpp_long_multiple, 378
- test_py_list_bytes_to_list_like_vector_char_multiple, 378
- test_py_list_bytes_to_list_vector_char_multiple, 379
- test_py_list_bytes_to_vector_vector_char_multiple, 379
- test_py_list_str16_to_list_like_u16string_multiple, 379
- test_py_list_str16_to_list_u16string_multiple, 379
- test_py_list_str16_to_vector_u16string_multiple, 379
- test_py_list_str32_to_list_like_u32string_multiple, 380
- test_py_list_str32_to_list_u32string_multiple, 380
- test_py_list_str32_to_vector_u32string_multiple, 380
- test_py_list_str_to_list_like_string_multiple, 380
- test_py_list_str_to_list_string_multiple, 380
- test_py_list_str_to_vector_string_multiple, 381
- test_py_list_to_list_like_multiple, 381
- test_py_list_to_list_multiple, 381
- test_py_list_to_vector_multiple, 381
- test_py_set_bytes_to_unordered_set_vector_char_multiple, 381
- test_py_set_str16_to_unordered_set_u16string_multiple, 382
- test_py_set_str32_to_unordered_set_u32string_multiple, 382
- test_py_set_str_to_unordered_set_string_multiple, 382
- test_py_set_to_unordered_set_multiple, 382
- test_py_str16_to_cpp_u16string_multiple, 382
- test_py_str32_to_cpp_u32string_multiple, 383
- test_py_str_to_cpp_string_multiple, 383
- test_py_tuple_bytes_to_list_like_vector_char_multiple, 383
- test_py_tuple_bytes_to_list_vector_char_multiple, 383
- test_py_tuple_bytes_to_vector_vector_char_multiple, 383
- test_py_tuple_str_to_list_like_string_multiple, 384
- test_py_tuple_str_to_list_string_multiple, 384
- test_py_tuple_str_to_vector_string_multiple, 384
- test_py_tuple_to_list_like_multiple, 384
- test_py_tuple_to_list_multiple, 384
- test_py_tuple_to_vector_multiple, 385
- TEST_REPEAT, 389
- test_unordered_set_string_to_py_set_multiple, 385
- test_unordered_set_to_py_set_multiple, 385
- test_unordered_set_u16string_to_py_set_multiple, 385
- test_unordered_set_u32string_to_py_set_multiple, 385
- test_unordered_set_vector_char_to_py_set_multiple, 386
- test_vector_string_to_py_list_multiple, 386
- test_vector_string_to_py_tuple_multiple, 386
- test_vector_to_py_list_multiple, 386
- test_vector_to_py_tuple_multiple, 386
- test_vector_u16string_to_py_list_multiple, 387
- test_vector_u32string_to_py_list_multiple, 387
- test_vector_vector_char_to_py_list_multiple, 387
- test_vector_vector_char_to_py_tuple_multiple, 387
- test_performance.h
 - test_performance_all, 389
- TEST_PERFORMANCE_ALL
 - main.cpp, 415
- test_performance_all
 - test_performance.cpp, 376
 - test_performance.h, 389
- TEST_PERFORMANCE_DICTS
 - test_performance.cpp, 357
- TEST_PERFORMANCE_FUNDAMENTAL_TYPES
 - test_performance.cpp, 358
- TEST_PERFORMANCE_LISTS
 - test_performance.cpp, 358

- TEST_PERFORMANCE_OBJECT_BOOL
 - test_performance.cpp, [358](#)
- TEST_PERFORMANCE_OBJECT_BYTES
 - test_performance.cpp, [358](#)
- TEST_PERFORMANCE_OBJECT_COMPLEX
 - test_performance.cpp, [358](#)
- TEST_PERFORMANCE_OBJECT_DOUBLE
 - test_performance.cpp, [358](#)
- TEST_PERFORMANCE_OBJECT_LONG
 - test_performance.cpp, [358](#)
- TEST_PERFORMANCE_OBJECT_STRING
 - test_performance.cpp, [358](#)
- TEST_PERFORMANCE_OBJECT_STRING_16
 - test_performance.cpp, [359](#)
- TEST_PERFORMANCE_OBJECT_STRING_32
 - test_performance.cpp, [359](#)
- TEST_PERFORMANCE_SETS
 - test_performance.cpp, [359](#)
- TEST_PERFORMANCE_TUPLES
 - test_performance.cpp, [359](#)
- test_py_bool_to_cpp_bool_multiple
 - test_performance.cpp, [376](#)
- test_py_bytes_to_cpp_vector_char_multiple
 - test_performance.cpp, [376](#)
- test_py_complex_to_cpp_complex_multiple
 - test_performance.cpp, [376](#)
- test_py_dict_to_cpp_std_map
 - test_common.h, [325](#)
- test_py_dict_to_cpp_std_map_bytes
 - test_common.cpp, [290](#)
 - test_common.h, [325](#)
- test_py_dict_to_cpp_std_map_like
 - test_common.h, [325](#)
- test_py_dict_to_cpp_std_map_like_bytes
 - test_common.cpp, [290](#)
- test_py_dict_to_cpp_std_map_like_multiple
 - test_performance.cpp, [376](#)
- test_py_dict_to_cpp_std_map_like_string
 - test_common.cpp, [290](#)
- test_py_dict_to_cpp_std_map_like_string16
 - test_common.cpp, [290](#)
- test_py_dict_to_cpp_std_map_like_string32
 - test_common.cpp, [291](#)
- test_py_dict_to_cpp_std_map_like_string_multiple
 - test_performance.cpp, [376](#)
- test_py_dict_to_cpp_std_map_like_vector_char_multiple
 - test_performance.cpp, [377](#)
- test_py_dict_to_cpp_std_map_multiple
 - test_performance.cpp, [377](#)
- test_py_dict_to_cpp_std_map_string
 - test_common.cpp, [291](#)
 - test_common.h, [325](#)
- test_py_dict_to_cpp_std_map_string16
 - test_common.cpp, [291](#)
 - test_common.h, [325](#)
- test_py_dict_to_cpp_std_map_string32
 - test_common.cpp, [291](#)
 - test_common.h, [326](#)
- test_py_dict_to_cpp_std_map_string_multiple
 - test_performance.cpp, [377](#)
- test_py_dict_to_cpp_std_map_vector_char_multiple
 - test_performance.cpp, [377](#)
- test_py_dict_to_cpp_std_unordered_map
 - test_common.h, [326](#)
- test_py_dict_to_cpp_std_unordered_map_bytes
 - test_common.cpp, [291](#)
 - test_common.h, [326](#)
- test_py_dict_to_cpp_std_unordered_map_multiple
 - test_performance.cpp, [377](#)
- test_py_dict_to_cpp_std_unordered_map_string
 - test_common.cpp, [292](#)
 - test_common.h, [326](#)
- test_py_dict_to_cpp_std_unordered_map_string_multiple
 - test_performance.cpp, [378](#)
- test_py_dict_to_cpp_std_unordered_map_u16string
 - test_common.cpp, [292](#)
 - test_common.h, [326](#)
- test_py_dict_to_cpp_std_unordered_map_u32string
 - test_common.cpp, [292](#)
 - test_common.h, [327](#)
- test_py_dict_to_cpp_std_unordered_map_vector_char_multiple
 - test_performance.cpp, [378](#)
- test_py_float_to_cpp_double_multiple
 - test_performance.cpp, [378](#)
- test_py_int_to_cpp_long_multiple
 - test_performance.cpp, [378](#)
- test_py_list_bytes_to_list_like_vector_char_multiple
 - test_performance.cpp, [378](#)
- test_py_list_bytes_to_list_vector_char_multiple
 - test_performance.cpp, [379](#)
- test_py_list_bytes_to_vector
 - test_common.cpp, [292](#)
 - test_common.h, [327](#)
- test_py_list_bytes_to_vector_vector_char_multiple
 - test_performance.cpp, [379](#)
- test_py_list_str16_to_list_like_u16string_multiple
 - test_performance.cpp, [379](#)
- test_py_list_str16_to_list_u16string_multiple
 - test_performance.cpp, [379](#)
- test_py_list_str16_to_vector
 - test_common.cpp, [293](#)
 - test_common.h, [327](#)
- test_py_list_str16_to_vector_u16string_multiple
 - test_performance.cpp, [379](#)
- test_py_list_str32_to_list_like_u32string_multiple
 - test_performance.cpp, [380](#)
- test_py_list_str32_to_list_u32string_multiple
 - test_performance.cpp, [380](#)
- test_py_list_str32_to_vector
 - test_common.cpp, [293](#)
 - test_common.h, [328](#)
- test_py_list_str32_to_vector_u32string_multiple
 - test_performance.cpp, [380](#)
- test_py_list_str_to_list_like_string_multiple
 - test_performance.cpp, [380](#)
- test_py_list_str_to_list_string_multiple

- test_performance.cpp, 380
- test_py_list_str_to_vector
 - test_common.cpp, 293
 - test_common.h, 328
- test_py_list_str_to_vector_string_multiple
 - test_performance.cpp, 381
- test_py_list_to_list_like_multiple
 - test_performance.cpp, 381
- test_py_list_to_list_multiple
 - test_performance.cpp, 381
- test_py_list_to_vector
 - test_common.h, 329
- test_py_list_to_vector_multiple
 - test_performance.cpp, 381
- test_py_list_to_vector_round_trip
 - test_common.h, 329
- test_py_set_bytes_to_unordered_set
 - test_common.cpp, 294
 - test_common.h, 329
- test_py_set_bytes_to_unordered_set_vector_char_multiple
 - test_performance.cpp, 381
- test_py_set_str16_to_unordered_set_u16string_multiple
 - test_performance.cpp, 382
- test_py_set_str32_to_unordered_set_u32string_multiple
 - test_performance.cpp, 382
- test_py_set_str_to_unordered_set_string_multiple
 - test_performance.cpp, 382
- test_py_set_string16_to_unordered_set
 - test_common.cpp, 294
 - test_common.h, 329
- test_py_set_string32_to_unordered_set
 - test_common.cpp, 294
 - test_common.h, 329
- test_py_set_string_to_unordered_set
 - test_common.cpp, 294
 - test_common.h, 329
- test_py_set_to_unordered_set
 - test_common.h, 330
- test_py_set_to_unordered_set_multiple
 - test_performance.cpp, 382
- test_py_str16_to_cpp_u16string_multiple
 - test_performance.cpp, 382
- test_py_str32_to_cpp_u32string_multiple
 - test_performance.cpp, 383
- test_py_str_to_cpp_string_multiple
 - test_performance.cpp, 383
- test_py_tuple_bytes_to_list_like_vector_char_multiple
 - test_performance.cpp, 383
- test_py_tuple_bytes_to_list_vector_char_multiple
 - test_performance.cpp, 383
- test_py_tuple_bytes_to_vector
 - test_common.cpp, 295
 - test_common.h, 330
- test_py_tuple_bytes_to_vector_vector_char_multiple
 - test_performance.cpp, 383
- test_py_tuple_str16_to_vector
 - test_common.cpp, 295
 - test_common.h, 330
- test_py_tuple_str32_to_vector
 - test_common.cpp, 295
 - test_common.h, 331
- test_py_tuple_str_to_list_like_string_multiple
 - test_performance.cpp, 384
- test_py_tuple_str_to_list_string_multiple
 - test_performance.cpp, 384
- test_py_tuple_str_to_vector
 - test_common.cpp, 296
 - test_common.h, 331
- test_py_tuple_str_to_vector_string_multiple
 - test_performance.cpp, 384
- test_py_tuple_to_list_like_multiple
 - test_performance.cpp, 384
- test_py_tuple_to_list_multiple
 - test_performance.cpp, 384
- test_py_tuple_to_vector
 - test_common.h, 332
- test_py_tuple_to_vector_multiple
 - test_performance.cpp, 385
- test_py_tuple_to_vector_round_trip
 - test_common.h, 332
- TEST_REPEAT
 - test_performance.cpp, 389
- test_unordered_set_bytes_to_py_set
 - test_common.cpp, 296
 - test_common.h, 333
- test_unordered_set_string_to_py_set
 - test_common.cpp, 296
 - test_common.h, 333
- test_unordered_set_string_to_py_set_multiple
 - test_performance.cpp, 385
- test_unordered_set_to_py_set
 - test_common.h, 333
- test_unordered_set_to_py_set_multiple
 - test_performance.cpp, 385
- test_unordered_set_u16string_to_py_set
 - test_common.cpp, 297
 - test_common.h, 333
- test_unordered_set_u16string_to_py_set_multiple
 - test_performance.cpp, 385
- test_unordered_set_u32string_to_py_set
 - test_common.cpp, 297
 - test_common.h, 333
- test_unordered_set_u32string_to_py_set_multiple
 - test_performance.cpp, 385
- test_unordered_set_vector_char_to_py_set_multiple
 - test_performance.cpp, 386
- test_vector_string_to_py_list
 - test_common.cpp, 297
 - test_common.h, 334
- test_vector_string_to_py_list_multiple
 - test_performance.cpp, 386
- test_vector_string_to_py_tuple
 - test_common.cpp, 297
 - test_common.h, 334
- test_vector_string_to_py_tuple_multiple
 - test_performance.cpp, 386

- test_vector_to_py_list
 - test_common.h, 334
- test_vector_to_py_list_multiple
 - test_performance.cpp, 386
- test_vector_to_py_list_round_trip
 - test_common.h, 335
- test_vector_to_py_tuple
 - test_common.h, 335
- test_vector_to_py_tuple_multiple
 - test_performance.cpp, 386
- test_vector_to_py_tuple_round_trip
 - test_common.h, 336
- test_vector_u16string_to_py_list
 - test_common.cpp, 298
 - test_common.h, 336
- test_vector_u16string_to_py_list_multiple
 - test_performance.cpp, 387
- test_vector_u16string_to_py_tuple
 - test_common.cpp, 298
 - test_common.h, 337
- test_vector_u32string_to_py_list
 - test_common.cpp, 299
 - test_common.h, 337
- test_vector_u32string_to_py_list_multiple
 - test_performance.cpp, 387
- test_vector_u32string_to_py_tuple
 - test_common.cpp, 299
 - test_common.h, 338
- test_vector_vector_char_to_py_list
 - test_common.cpp, 300
 - test_common.h, 338
- test_vector_vector_char_to_py_list_multiple
 - test_performance.cpp, 387
- test_vector_vector_char_to_py_tuple
 - test_common.cpp, 300
 - test_common.h, 338
- test_vector_vector_char_to_py_tuple_multiple
 - test_performance.cpp, 387
- testCount
 - TestResult, 226
- TestFramework.cpp
 - count_of_unique_string, 237
 - operator<<, 237, 239
 - REGEX_SPACE_ANYTHING, 236
 - REGEX_SPACE_FLOAT, 237
 - REGEX_SPACE_INTEGER, 237
 - REGEX_SPACE_STRING_NO_SPACE, 237
 - reset_count_of_unique_string, 239
 - str_count, 240
 - TIME_PRECISION, 241
 - TIME_WIDTH, 241
 - unique_string, 239
 - unique_u16string, 239
 - unique_u32string, 240
 - unique_vector_char, 240
- TestFramework.h
 - count_of_unique_string, 242
 - operator<<, 242
 - reset_count_of_unique_string, 243
 - unique_string, 243
 - unique_u16string, 243
 - unique_u32string, 243
 - unique_vector_char, 244
- TestResult, 220
 - atomicTestMeanExecTime, 224
 - execTime, 224
 - execTimeAdd, 224
 - execTimeMax, 224
 - execTimeMin, 225
 - execTimeStdDev, 225
 - failed, 225
 - hasExecTimeStdDev, 225
 - name, 225
 - numScaleValues, 225
 - numTests, 226
 - operator=, 226
 - scaleValues, 226
 - setFailed, 226
 - testCount, 226
 - TestResult, 223, 224
 - totalTime, 226
- TestResultS, 227
 - dump_header, 228
 - dump_tail, 228
 - dump_tests, 228
 - failed, 228
 - push_back, 228
 - results, 228
 - TestResultS, 227
 - tResults, 227
- tHiResDouble
 - ExecClock, 212
- TIME_PRECISION
 - TestFramework.cpp, 241
- TIME_WIDTH
 - TestFramework.cpp, 241
- totalTime
 - TestResult, 226
- tResults
 - TestResultS, 227
- tuple_reverse
 - cPyCppContainers.cpp, 406
- UNARY_COLLECTIONS
 - src.py.code_gen, 202
- unary_declarations
 - src.py.code_gen, 198
- unary_definitions
 - src.py.code_gen, 198
- unique_string
 - TestFramework.cpp, 239
 - TestFramework.h, 243
- unique_u16string
 - TestFramework.cpp, 239
 - TestFramework.h, 243
- unique_u32string
 - TestFramework.cpp, 240

- TestFramework.h, [243](#)
- unique_vector_char
 - TestFramework.cpp, [240](#)
 - TestFramework.h, [244](#)
- vector_double_x2
 - cPyCppContainers.cpp, [407](#)
- very_generic_cpp_std_list_like_to_py_unary
 - Python_Cpp_Containers, [193](#)
- very_generic_py_unary_to_cpp_std_list_like
 - Python_Cpp_Containers, [194](#)
- WIDTH
 - src.py.code_gen_documentation, [208](#)
- write_files
 - src.py.code_gen, [198](#)