50 Boost C++ Libraries in 180 minutes

- What has Boost to offer?
- How do the Boost libraries look like?
- Which Boost library shall I use for a certain task?
- Which Boost libraries can I ignore?
- Where do I find more information?

Boris Schäling, boris@highscore.de

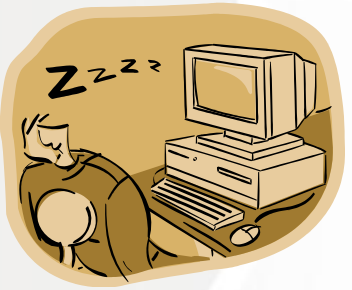
C++Now!, Aspen, 14 May 2012

Which 50 libraries?

General purpose libraries

Libraries which are useful for many developers and for the development of many programs

No TR1/C++11 libraries



Libraries in the C++ standard well-known by many developers

No „deprecated“ Boost libraries

A few Boost libraries have been superseded by newer versions or C++11

TR1/C++11 libraries

A lot of functionality of the Boost libraries is available through the standard library:

- Boost.Array
- Boost.Bind
- Boost.Chrono
- Boost.Function
- Boost.Hash
- Boost.Math
- Boost.MemberFunction
- Boost.Random
- Boost.Ref
- Boost.Regex
- Boost.SmartPointers
- Boost.System
- Boost.Thread
- Boost.Tuple
- Boost.TypeTraits
- Boost.Unordered

"Deprecated" libraries

Boost.Signals

Replaced by thread-safe Boost.Signals2 (thread-safety can be disabled)

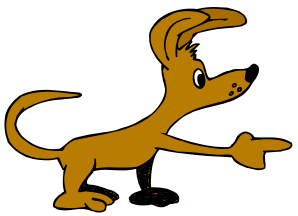
Boost.Lambda

Domain-specific language which looks like C++ but isn't C++; use for very small lambda functions or use C++11

Boost.Foreach

Macro which simulates foreach-loops from other programming languages; use for-loop or C++11

Resource management

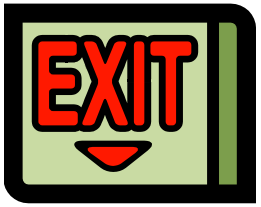


Boost.SmartPointers

Managing a dynamically allocated object in a shared or in a scoped (unique) pointer

Boost.PointerContainer

Managing many dynamically allocated objects which are owned by a container exclusively



Boost.ScopeExit

A macro to clean up ressources without using pointers

Resource management

Boost.Pool

Memory management with an allocator based on a singleton

Boost.CompressedPair

boost::compressed_pair<> like std::pair<> with empty base class optimization

Boost.SmartPointers

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/shared_ptr.hpp>
#include <boost/scoped_array.hpp>

using namespace boost;

shared_ptr<std::string> sp(new std::string("Hello, world!"));
scoped_array<std::string> sa(new std::string[100]);

#include <boost/make_shared.hpp>

shared_ptr<int> sp = make_shared<int>(99);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.SmartPointers

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

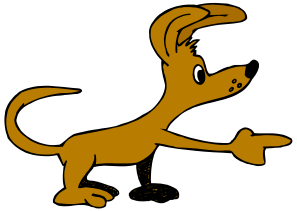
```
#define BOOST_SP_DISABLE_THREADS
#define BOOST_SP_USE_QUICK_ALLOCATOR
#include <boost/shared_ptr.hpp>

using namespace boost;

HANDLE winHandle = ...;
shared_ptr<void> sp(winHandle, CloseHandle);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.SmartPointers

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/intrusive_ptr.hpp>

using namespace boost;

void intrusive_ptr_add_ref(IUnknown *p) { p->AddRef(); }
void intrusive_ptr_release(IUnknown *p) { p->Release(); }

IUnknown *p = /* Microsoft COM object */;
intrusive_ptr<IUnknown> ip(p, false);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.PointerContainer

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

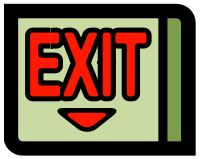
```
#include <boost/ptr_container/ptr_vector.hpp>

using namespace boost;

ptr_vector<int> v;
v.push_back(new int(1));
v.push_back(new int(2));

#include <boost/ptr_container/ptr_set.hpp>

ptr_set<int> s;
s.insert(new int(1));
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.ScopeExit

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/scope_exit.hpp>

void foo() {
  int *i = new int(99);
  BOOST_SCOPE_EXIT((i)) {
    delete i;
  } BOOST_SCOPE_EXIT_END
  *i = 100;
  i = nullptr;
}
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Pool

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/pool/pool_alloc.hpp>
#include <boost/pool/singleton_pool.hpp>

using namespace boost;
{
  std::vector<int, pool_allocator<int>> v;
  for (int i = 0; i < 1000000; ++i)
    v.push_back(i);
}

singleton_pool<pool_allocator_tag, sizeof(int)>::
  release_memory();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

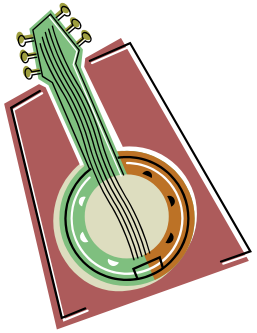
Something with strings

Boost.StringAlgorithms

Lots of free-standing functions to compare, find, replace, erase, trim … strings

Boost.Spirit

Parsing complex data formats using EBNF as a domain-specific language



Boost.Format

Writing strings with a type-safe and extensible sprintf()-like function

Boost.StringAlgorithms

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/algorithm/string/regex.hpp>

using namespace boost::algorithm;

std::string s = "Hello, world!";
std::cout << to_upper_copy(s) << std::endl;
std::cout << erase_all_copy(s, "l") << std::endl;
std::cout << replace_first_copy(s, "o", "ooo") << std::endl;
std::cout << trim_copy_if(s, is_any_of("!")) << std::endl;
to_lower(s);
std::cout << s << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Spirit

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/spirit/include/qi.hpp>

using namespace boost::spirit::qi;

template <typename Iterator>
struct foo : grammar<Iterator>
{
  foo() : foo::base_type(obj) { obj = +int_ >> "foo"; }
  rule<Iterator> obj;
};

foo<std::string::iterator> f;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Spirit

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
std::string s = "123foo";
bool b = parse(s.begin(), s.end(), f);
std::cout << b << std::endl; // 1
// -----
template <typename Iterator, typename Skipper>
struct foo : grammar<Iterator, Skipper> ...

foo<std::string::iterator, ascii::space_type> f;
s = "123 foo";
bool b = parse_phrase(s.begin(), s.end(), f, ascii::space);
std::cout << b << std::endl; // 1
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Format

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/format.hpp>

using namespace boost;

std::cout << format("%d %s") % 1 % "Hello!" << std::endl;
std::cout << format("%s %d") % 1 % "Hello!" << std::endl;
std::cout << format("%2% %1%") % 1 % "Hello!" << std::endl;
std::cout << format("%+d %10s") % 1 % "Hello!" << std::endl;
std::cout << format("%10s %+d") % 1 % "Hello!" << std::endl;
std::cout << format("%|2$10| %|1$+|") % 1 % "Hello!"
  << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Regular expressions

Boost.Regex

The C++ regular expression library which became a part of the standard with C++11

Boost.Xpressive

Basically a copy of Boost.Regex to write regular expressions as C++ code (a bit like Boost.Spirit)

Boost.StringAlgorithms

String processing functions based on the class boost::regex from Boost.Regex

Boost.Regex

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/regex.hpp>

using namespace boost;

std::string s = "Hello, world!";
regex re("\\w+,\\s\\w+!");
bool b = regex_match(s, re);
std::cout << b << std::endl; // 1
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Xpressive

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/xpressive/xpressive.hpp>

using namespace boost::xpressive;

std::string s = "Hello, world!";
sregex re = +_w >> "," >> _s >> +_w >> "!";
bool b = regex_match(s, re);
std::cout << b << std::endl; // 1
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.StringAlgorithms

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/algorithm/string/regex.hpp>

using namespace boost::algorithm;

std::string s = "Hello, world!";
boost::regex re("\\w+,\\s\\w+!");
boost::iterator_range<std::string::iterator> r =
  find_regex(s, re);
bool b = (r.begin() != r.end());
std::cout << b << std::endl; // 1
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Tokenizers

Boost.Tokenizer

Container with a TokenizerFunction concept and a few implementations

Boost.StringAlgorithms

Splitting a string with a function object and putting tokens into a container

Boost.Regex (and Boost.Xpressive)

An iterator which returns a token based on a regular expression

Boost.Tokenizer

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/tokenizer.hpp>

using namespace boost;

typedef tokenizer<escaped_list_separator<char>> tokenizer;
std::string s = "\"Hello, world!\"";
tokenizer tok(s);
for (std::string t : tok)
  std::cout << t << std::endl; // Hello, world!
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.StringAlgorithms

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/algorithm/string.hpp>

using namespace boost::algorithm;

std::string s = "Hello, world!";
std::vector<std::string> v;
split(v, s, is_space());
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Regex

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/regex.hpp>

using namespace boost;

std::string s = "Hello, world!";
regex re("\\w+");
regex_token_iterator<std::string::iterator> it(
  s.begin(), s.end(), re);
regex_token_iterator<std::string::iterator> end;
while (it != end)
  std::cout << *it++ << std::endl; // Hello
                                   // world
```

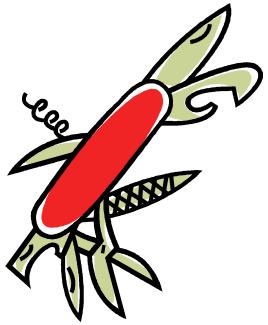
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Containers

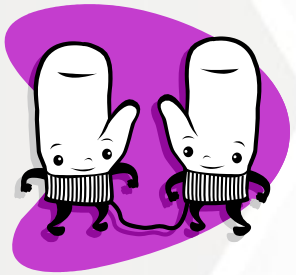


Boost.Multiindex

Create new containers which provide multiple interfaces to lookup items

Boost.Bimap

A ready-to-use container based on Boost.Multiindex with exactly two interfaces



Boost.CircularBuffer

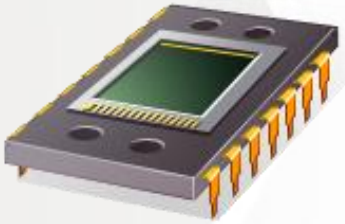
A fixed-size container which overwrites items if you keep on inserting more

Containers

Boost.MultiArray

Arrays with multiple dimensions (compile time) and arbitrarily long dimensions (run-time)

Boost.DynamicBitset



Works exactly like std::bitset except that the size can be set (and modified) at run-time

Boost.PropertyTree

A tree container with key/value pairs which can be saved to and loaded from files

Containers

Boost.Intrusive

Containers which don‘t allocate memory, copy no values and don‘t throw exceptions

Boost.Container

Same containers as in C++11 and a few more with move semantics

Boost.Heap

A priority queue like std::priority_queue but with more functionality

Boost.Multiindex

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/multi_index_container.hpp>

using namespace boost::multi_index;

typedef multi_index_container<
 person,
 indexed_by<
  hashed_unique<member<person, int, &person::id>>,
  hashed_non_unique<member<person, std::string, &person::name>>,
  ordered_non_unique<const_mem_fun<person, int, &person::age>>
 >
> person_multi;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Multiindex

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
person_multi ps;

ps.insert(person(1, "Alice", 40));
ps.insert(person(2, "Bob", 50));
ps.insert(person(3, "Bob", 60));

std::cout << ps.find(1)->name << std::endl; // Alice

const person_multi::nth_index<1>::type &name_index =
  ps.get<1>();
std::cout << name_index.count("Bob") << std::endl; // 2
```

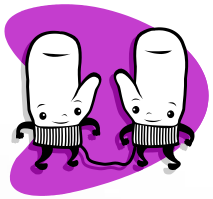
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Bimap



50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/bimap.hpp>
#include <boost/bimap/suppport/lambda.hpp>
using namespace boost::bimaps;

typedef bimap<int, multiset_of<double>> bimap;

bimap bm;
bm.insert(bimap::value_type(0, 0.1));
bm.insert(bimap::value_type(1, 0.1));

std::cout << bm.right.count(0.1) << std::endl;
bm.left.modify_key(bm.left.find(0), _key = 2);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.CircularBuffer

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/circular_buffer.hpp>

using namespace boost;

typedef circular_buffer<int> circular_buffer;
circular_buffer cb(3, 99);

cb.push_back(100);
for (int i : cb)
  std::cout << i << std::endl;

cb.linearize();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.MultiArray

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/multi_array.hpp>

using namespace boost;

multi_array<char, 2> a(extents[2][7]);
multi_array<char, 2>::reference subarray = a[0];
std::memcpy(subarray.origin(), "Hello, ", 7);
typedef multi_array<char, 2>::array_view<1>::type array_view;
typedef multi_array<char, 2>::index_range range;
array_view view = a[indices[1][range(0, 6)]];
std::memcpy(subarray.origin(), "world!", 6);
```

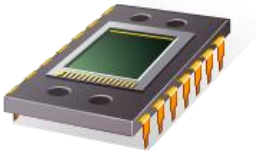
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.DynamicBitset



50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/dynamic_bitset.hpp>

using namespace boost;

dynamic_bitset<> db(3, 4);
db.push_back(true);
std::cout << db.size() << std::endl;
std::cout << db.count() << std::endl;
std::cout << db.any() << std::endl;
std::cout << db[0].flip() << std::endl;
std::cout << ~db[0] << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.PropertyTree

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/property_tree/ptree.hpp>

using namespace boost::property_tree;

ptree pt;
pt.put("C:.Windows", "10");
pt.put("C:.Windows.System", "20");
std::cout << pt.get<int>(ptree::path_type("C:\\Windows", '\\'))
  + pt.get<int>("C:.Windows.System") << std::endl; // 30
for (auto a : pt.get_child("C:"))
  std::cout << a.first << std::endl; // Windows
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.PropertyTree

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/property_tree/json_parser.hpp>

iptree pt;
pt.put("C:.WINDOWS", "10");
pt.put("C:.Windows.System", "20");
json_parser::write_json("file.json", pt);

ptree pt2;
json_parser::read_json("file.json", pt);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Intrusive

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/intrusive/list.hpp>

using namespace boost::intrusive;

struct foo : public list_base_hook<> {};

typedef list<foo> foo_list;

foo_list l;
foo f;
l.push_back(f);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Intrusive

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
foo_list l;
foo *f = new foo();
l.push_back(*f);
l.pop_back_and_dispose([](foo *f){ delete f; });
// -----
struct foo : public list_base_hook<link_mode<auto_unlink>> {};

typedef list<foo, constant_time_size<false>> foo_list;

foo_list l;
{ foo f; l.push_back(f); }
std::cout << l.empty() << std::endl; // 1
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

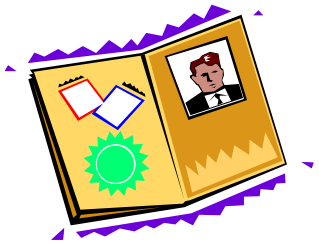
Data structures

Boost.Any

A type which makes variables behave like typeless variables (eg. like in Javascript)

Boost.Variant

Similar to Boost.Any but with a restricted set of types



Boost.Uuid

Create universally unique identifiers (like the ones used by Microsoft COM)

Data structures

Boost.Optional

Makes it possible to set a variable to NULL even if it‘s not a pointer

Boost.Tribool

Like bool but with a third possible state of indeterminate

Boost.Any

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/any.hpp>

using namespace boost;

any a = 1;
a = 3.14;
a = true;
a = std::string("Hello, world!");
if (!a.empty())
  std::cout << any_cast<std::string>(a) << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Variant

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/variant.hpp>

using namespace boost;

variant<double, char, std::string> v;
v = 3.14;
v = 'A';
v = "Hello, world!";
std::cout << get<std::string>(v) << std::endl;
std::cout << v << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Variant

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

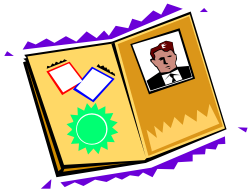
```
#include <boost/variant.hpp>

using namespace boost;

struct visitor : public static_visitor<>
{
  template <typename T>
  void operator()(T &t) const { std::cout << t << std::endl; }
}

variant<double, char, std::string> v;
apply_visitor(visitor(), v);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.Uuid

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/uuid/uuid.hpp>
#include <boost/uuid/uuid_generators.hpp>

using namespace boost::uuids;

random_generator gen;
uuid id = gen();
std::cout << id << std::endl;
string_generator gen2;
id = gen2("1c7ccba0-1376-4366-bc80-034e6a1191f0");
std::string s = id.to_string();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Optional

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/optional.hpp>

using namespace boost;

optional<int> i = 1;
i = none;
if (i)
  std::cout << *i << std::endl;
if (i.is_initialized())
  std::cout << i.get() << std::endl;
std::cout << get_optional_value_or(i, 0) << std::endl; // 0
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Tribool

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/logic/tribool.hpp>

using namespace boost::logic;

tribool b;
b = true;
b = false;
b = indeterminate;

if (b) ...
else if (!b) ...
else ...
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

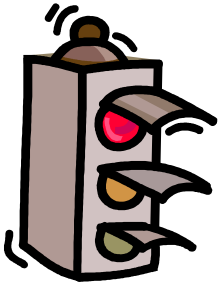
Design patterns

Boost.Flyweight

Flyweight pattern: Sharing common data between objects to minimize memory usage

Boost.Asio

Reactor pattern: Demultiplexing requests and dispatching them to request handlers



Boost.Signals2

Observer pattern: Notifying observers about state changes in a subject

Boost.Flyweight

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/flyweight.hpp>

using namespace boost::flyweights;

std::vector<flyweight<std::string>> v;
for (int i = 0; i < 1000; ++i)
  v.push_back("Hello, world!");
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Asio

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

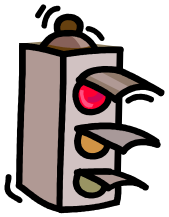
```
#include <boost/asio.hpp>

using namespace boost::asio;

io_service ios;
deadline_timer timer1(ios, boost::posix_time::seconds(1));
timer1.async_wait([](boost::system::error_code ec){});
deadline_timer timer2(ios, boost::posix_time::seconds(2));
timer2.async_wait([](boost::system::error_code ec){});
ios.run();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.Signals2

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/signals2.hpp>

using namespace boost::signals2;

signal<void()> s;
s.connect([](){ std::cout << "Hello, "; });
s.connect([](){ std::cout << "world!"; });
s();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

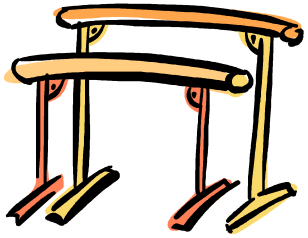
Communication

Boost.Asio

Asynchronous I/O for network programming and OS-specific operations

Boost.Interprocess

Creating and accessing shared memory to communicate with other processes



Boost.MPI

A runtime environment for parallel computing with multiple instances of a program

Boost.Asio

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/asio.hpp>

using namespace boost::asio;

io_service ios;
ip::tcp::endpoint endpoint(ip::tcp::v4(), 1234);
ip::tcp::acceptor acceptor(ios, endpoint);
ip::tcp::socket sock(ios);
acceptor.listen();
acceptor.async_accept(sock, [](boost::system::error_code&){});
ios.run();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Asio

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/asio/windows/object_handle.hpp>

using namespace boost::asio;

io_service ios;
PROCESS_INFORMATION pi;
CreateProcess("program.exe", ..., &pi);
windows::object_handle handle(ios, pi.hProcess);
handle.async_wait([](boost::system::error_code ec){});
ios.run();
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Interprocess

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

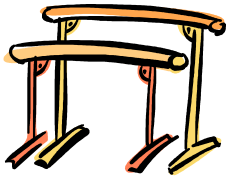
```
#include <boost/interprocess/managed_shared_memory.hpp>

using namespace boost::interprocess;

managed_shared_memory shmem(open_or_create, "name", 1024);
shmem.construct<int>("age")(99);
std::pair<int*, std::size_t> p = shmem.find<int>("age");
shmem.destroy<int>("age");
typedef allocator<char, managed_shared_memory::segment_manager>
  CharAllocator;
typedef boost::interprocess::basic_string<char,
  std::char_traits<char>, CharAllocator> string;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.MPI

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

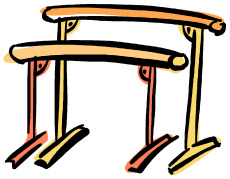
```
#include <boost/mpi.hpp>

using namespace boost::mpi;

environment env;
communicator world;
std::string s;
if (world.rank() == 0)
  s = "Hello, world!";
broadcast(world, s, 0);
// Run with: mpiexec -n 4 sample.exe
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.MPI

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
environment env;
communicator world;
std::vector<std::string> v =
  { "Hello, world!", "Hello, moon!", "Hello, sun!" };
std::string s;
scatter(world, v, s, 0);

std::string min(const std::string &lhs, const std::string &rhs) {
  return lhs.size() < rhs.size() ? lhs : rhs;
}
std::string result;
reduce(world, s, result, min, 0);
```

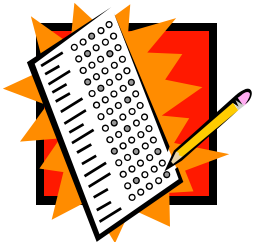
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Application development



Boost.ProgramOptions

Define command line options and evaluate command line arguments

Boost.Log

A logging library reviewed and accepted but not yet shipped with the Boost libraries

Boost.ProgramOptions

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/program_options.hpp>

using namespace boost::program_options;

options_description desc;
desc.add_options()("help", "Help screen");

variables_map vm;
store(parse_command_line(argc, argv, desc), vm);
notify(vm);

if (vm.count("help")) ...
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

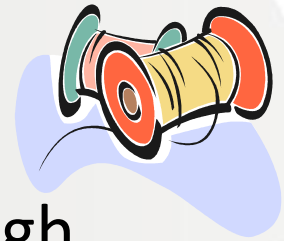
C++ now

System

Boost.Filesystem

Process paths and access the filesystem (not only files but also directories)

Boost.Thread



Create threads just like with C++11;
Boost.Thread has interruptable threads though

Boost.Filesystem

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/filesystem.hpp>

using namespace boost::filesystem;

path p(R"(C:\test)");
std::cout << p.generic_string() << std::endl; // C:/test
create_directory(p);
rename(p, R"(C:\test2)");
remove(R"(C:\test2)");
directory_iterator it(current_path());
while (it != directory_iterator())
  std::cout << *it++ << std::endl;
```

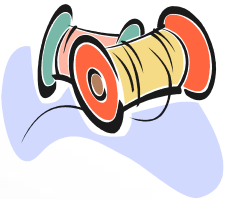
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Thread



50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/thread.hpp>

using namespace boost;

thread t([](){
  try {
    while (true) {
      this_thread::sleep(boost::posix_time::seconds(1));
    }
  } catch (thread_interrupted&) {}
});
t.interrupt();
```

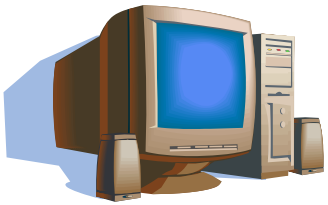
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Error handling

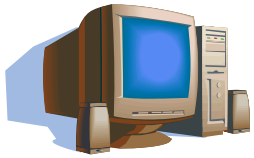


Boost.System

Four classes for error codes, error categories and errors as exceptions

Boost.Exception

An exception class information can be easily added to after it has been thrown



Boost.System

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/system/error_code.hpp>
#include <boost/system/system_error.hpp>

using namespace boost;

error_code ec = make_error_code(errc::too_many_file_open);
std::cout << ec.value() << std::endl; // 24
const error_category &cat = ec.category();
std::cout << cat.name() << std::endl; // generic
error_condition con = ec.default_error_condition();
std::cout << con.value() << std::endl; // 24
throw system_error(ec);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Exception

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/exception/all.hpp>

using namespace boost;

typedef error_info<struct tag_errmsg, std::string> errmsg_info;

try {
  throw exception();
}
catch (exception &ex) {
  ex << errmsg_info("Now I know why it failed");
}
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Exception

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
try {
  try {
    BOOST_THROW_EXCEPTION(user_defined_exception());
  }
  catch (exception &ex) {
    ex << errmsg_info("Now I know why it failed");
  }
}
catch (exception &ex) {
  diagnostic_information(ex);
  std::cout << get_error_info<errmsg_info>(ex) << std::endl;
}
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

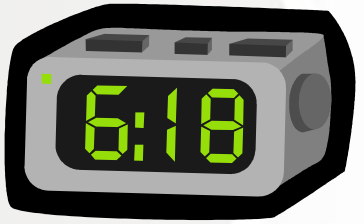
Time

Boost.DateTime

A library for calendar dates and times with extensive support for flexible input and output

Boost.Chrono

Provides a lot of clocks to measure wall clock time, process time, monotonic time ...



Boost.Timer

Based on a particular clock from Boost.Chrono to profile code

Boost.DateTime

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/date_time/gregorian/gregorian.hpp>

using namespace boost::gregorian;

date d(2012, 5, 14);
date_duration dd(31);
date d2 = d + dd;
date_period dp(d, d2);
day_iterator it(d);
std::cout << next_weekday(d, greg_weekday(date_time::Friday))
  << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Chrono

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/chrono.hpp>

using namespace boost::chrono;

std::cout << system_clock::now() << std::endl;
auto then = process_real_cpu_clock::now();
// do something
auto now = process_real_cpu_clock::now();
std::cout << now - then << std::endl;
std::cout << now + milliseconds(100) << std::endl;
std::cout << time_point_cast<minutes>(now) << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Timer

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/timer.hpp>

using namespace boost::timer;

cpu_timer timer;
// do something
timer.stop();
// do something
timer.resume();
// do something
std::cout << timer.format() << std::endl;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

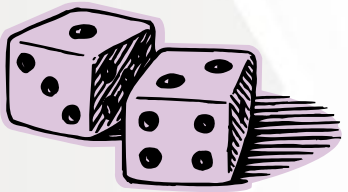
Math

Boost.Integer

Integer types with exact, minimum and fast sizes

Boost.Random

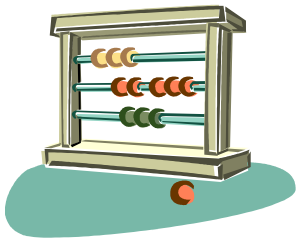
Random number generators with distributors to generate numbers with restrictions



Boost.Accumulators

Containers which calculate new results whenever a new value is pushed into them

Math



Boost.Rational

Use exact representations of rational numbers like 1/3 in C++

Boost.MathCommonFactor

Find the greatest common divisor and least common multiple

Boost.Graph

A library to solve problems like finding the shortest route between two subway stations

Boost.Integer

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/cstdint.hpp>

using namespace boost;

int8_t i8 = 1;
uint_least32_t uil32 = 1;
int_fast16_t if16 = 1;
#ifndef BOOST_NO_INT64_T
uint64_t ui64 = 1;
#endif
intmax_t imax = 1;
```

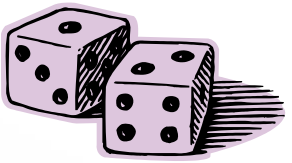
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Random



50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/random.hpp>

using namespace boost::random;

mt19937 gen(std::time(0));
std::cout << gen() << std::endl; // 2047385591
bernoulli_distribution<> dist;
std::cout << dist(gen) << std::endl; // 0
uniform_int_distribution<> dist2(1, 1000);
std::cout << dist2(gen) << std::endl; // 146
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Accumulators

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/accumulators/accumulators.hpp>
#include <boost/accumulators/statistics.hpp>

using namespace boost::accumulators;

accumulator_set<int, features<tag::count>> acc;
acc(1);
acc(-2);
std::cout << count(acc) << std::endl; // 2
acc(4);
std::cout << count(acc) << std::endl; // 3
```

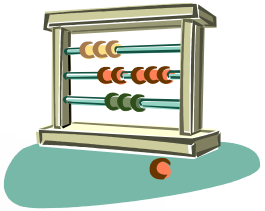
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Rational



50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/rational.hpp>

using namespace boost;

rational<int> r(1, 2);
std::cout << r << std::endl; // 1/2
std::cout << r + r << std::endl; // 1/1
std::cout << r * r << std::endl; // 1/4
std::cout << rational_cast<float>(r) << std::endl; // 0.5
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.MathCommonFactor

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/math/common_factor_rt.hpp>

using namespace boost::math;

std::cout << lcm(12, 16) << std::endl; // 48
std::cout << gcd(12, 16) << std::endl; // 4
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Graph

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/graph/adjacency_list.hpp>

using namespace boost;

enum { top, bottom };

std::array<std::pair<int, int>, 1> edges = {
  std::make_pair(top, bottom)
};

typedef adjacency_list<setS, vecS, undirectedS> graph;
graph g(edges.begin(), edges.end(), 1);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Graph

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
std::array<int, 2> distances = { 0 };

#include <boost/graph/breadth_first_search.hpp>
#include <boost/graph/named_function_params.hpp>
#include <boost/graph/visitors.hpp>

breadth_first_search(g, top,
  visitor(
    make_bfs_visitor(
      record_distances(distances.begin(),
        on_tree_edge()))));
```

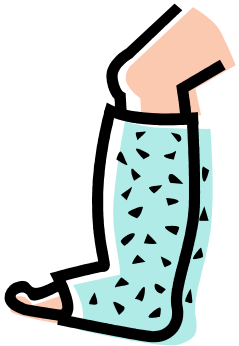
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Cast Operators



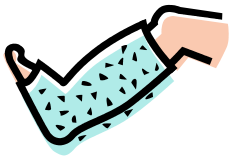
Boost.Conversion

Three cast operators for numbers and polymorphic types

Boost.NumericConversion

A cast operator to detect overflows when converting from big to small numeric types

More cast operators in Boost.Rational (rational_cast<>), Boost.Chrono (time_point_cast<>, duration_cast<>), Boost.Any (any_cast<>) …



Boost.Conversion

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

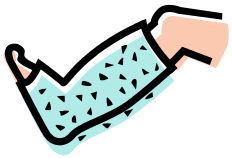
```
#include <boost/cast.hpp>

using namespace boost;

struct father { virtual ~father() {} };
struct mother { virtual ~mother() {} };
struct child : public father, public mother {};

father *f = new child();
child *c = polymorphic_downcast<child*>(f);
mother *m = polymorphic_cast<mother*>(f);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now



Boost.Conversion

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/cast.hpp>

using namespace boost;

std::string s = lexical_cast<std::string>(123);
int i = lexical_cast<int>(s);

try
{
  lexical_cast<int>("abc");
}
catch (bad_lexical_cast &e) {}
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.NumericConversion

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/numeric/conversion/cast.hpp>

using namespace boost;

try
{
  int i = 0x10000;
  short s = numeric_cast<short>(i);
}
catch (numeric::bad_numeric_cast &e) {}
```

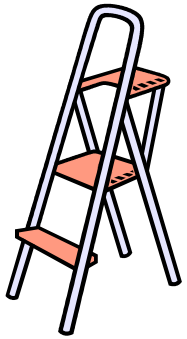
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Utilities

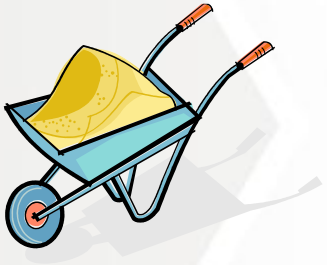


Boost.Utility

Small utilities which were too small for their own libraries

Boost.Assign

Initialize containers and add multiple values without calling push_back() dozens of times



Boost.StaticAssert

Check the size of a type at compile time

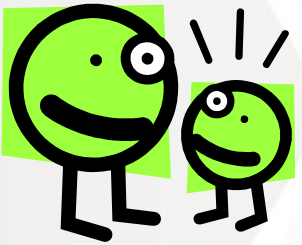
Utilities

Boost.Operators

Add operators to your class by deriving from helper classes which define them for you

Boost.MinMax

Find the minimum and maximum of two or multiple values with one function call



Boost.Swap

Like std::swap() but uses optimized swap implementations for many Boost libraries

Utilities

Boost.Hash

Classes and functions to return hash values and to build your own for user-defined types

There are many more small Boost libraries which could be put into this section of the presentation

Boost.Utility

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/next_prior.hpp>

array<int, 4> a = { 1, 2, 3, 4 };
std::cout << *next(a.begin()) << *prior(a.end()) << std::endl;

#include <boost/noncopyable.hpp>

struct x : boost::noncopyable {};

#include <boost/utility/binary.hpp>

std::cout << BOOST_BINARY(1010) << std::endl;
```

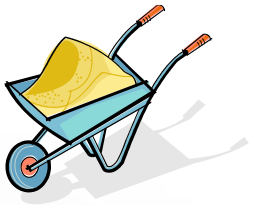
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Assign



50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/assign.hpp>

using namespace boost::assign;

std::vector<int> v = list_of(1)(2)(3);
std::map<std::string, int> m = map_list_of("a", 1)("b", 2);

push_back(v)(4)(5)(6);

#include <boost/assign/std/vector.hpp>

v += 7, 8, 9;
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.StaticAssert

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/static_assert.hpp>

#pragma pack(push, 1)

struct message {
  int16_t i;
  uint32_t ui;
};

BOOST_STATIC_ASSERT(sizeof(message) == 6);

#pragma pack(pop)
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Operators

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/operators.hpp>

using namespace boost;

struct foo : public equality_comparable<foo> {
  bool operator==(const foo &f) const {
    return true;
  }
};

foo f1, f2;
std::cout << (f1 != f2) << std::endl; // 0
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.MinMax

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/algorithm/minmax.hpp>

using namespace boost;

typedef std::array<int, 4> array;
array a = { 0, 1, 2, 3 };

boost::tuple<const int&, const int&> t = minmax(a[0], a[1]);

std::pair<array::iterator, array::iterator> p =
  minmax_element(a.begin(), a.end());
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Swap

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/swap.hpp>

boost::array<int, 2> a1 = { 1, 2 };
boost::array<int, 2> a2 = { 3, 4 };

boost::swap(a1, a2);
```

100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

Boost.Hash

50BoostLibraries - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

main.cpp

(Global Scope)

```
#include <boost/functional/hash.hpp>

using namespace boost;

struct foo { std::string s; int i; };

std::size_t hash_value(const foo &f) {
  std::size_t seed = 0;
  hash_combine(seed, f.s);
  hash_combine(seed, f.i);
  return seed;
}
```

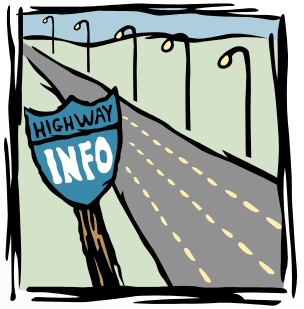
100 %

Error List Output Find Symbol Results

Ready Ln 1 Col 1 Ch 1 INS

C++ now

More information



- Boost documentation:

  http://www.boost.org/doc/libs
- Presentations from BoostCons:

  http://boostcon.boost.org/presentations/
- Online book:

  http://en.highscore.de/cpp/boost/

  http://www.highscore.de/cpp/boost/ (German)

  http://zh.highscore.de/cpp/boost/ (Chinese)
- References to blogs, books, articles:

  http://svn.boost.org/trac/boost/wiki/References