C++11 BLWS (Appendix)

The above topics will only be covered shortly if there is special request.

Major Topics Not Covered

1. C compatibility
2. I18n and L18n
3. Complex Numbers
4. Global Numeric Functions
5. Valarray
6. Allocators

This section is provided to be filled with optional print-outs from the Boost Documentation.

C compatibility

It is usually easy to access libraries written in C from C++.

> The other way round can be much harder!

The first step when trying to make a large C code base available in C++ should be the necessary clean-up so that the code compiles with both, a C and a C++ compiler.

(More of this topic can be covered on demand.)

I18n and L18n

Internationalisation and Localisation quickly turn into a huge topic, but – with respect to learning C++ – it is a more peripheral aspect of the language and library.

> For the relation to standard library functions see:
> http://en.cppreference.com/w/cpp/locale

Besides detailed control over formatting of numeric data also message catalogues are supported.*

Note that there are (still) no such things like standard message catalogues*

Also note with respect to character set conversion support is very basic and for heavily working with text-oriented applications to support foreign languages often the ICU Library is among the recommended solutions.

(More of this topic can be covered on demand.)

*: Similar to what is available with GNU `gettext` or POSIX `catgets`.

Complex Numbers

Complex Numbers are available in C++ via the `std::complex`-template. Implementations need (only) to support specialisations for `float`, `double`, and `long double`.

> For more information see:
> http://en.cppreference.com/w/cpp/numeric/complex

(More of this topic can be covered on demand.)

Common Mathematical Functions

The <cmath> header covers many standard numeric functions, *sine* and *cosine* among them as well as rounding and truncating conversions and more.

> For more information see:
> http://en.cppreference.com/w/cpp/numeric/math

(More of this topic can be covered on demand.)

Valarray

The `std::valarray` template was introduced in C++98 to support operations on array storage that models the mathematical of a vector. It did not gain too much (widespread) use, maybe due to the fact that there are some other popular alternatives freely available.

> For more information see:
> http://en.cppreference.com/w/cpp/numeric/valarray

(More of this topic can be covered on demand.)

Allocators

Allocators are not exactly a class (in the meaning of something put into a library and providing a certain service) but rather a customization point for the standard library containers.

> For more information see:
> http://en.cppreference.com/w/cpp/concept/Allocator

(More of this topic can be covered on demand.)

Installing Boost

This section is provided to be filled with optional print-outs
from the Boost Documentation.

How to get Boost

There are several ways to get Boost or parts of it:

For getting **all of** Boost start at http://www.boost.org/users/download/

- in pre-compiled form
  - for MS-Windows (look it up here)
  - for (various) Linux Variants (try locate it with your package manager)
- as source code (to build yourself)
  - for MS-Windows follow instructions here
  - for "Unix-like" Systems (of course including Linux) follow instructions here

Extracting Subsets

As many Boost (Sub-) Libraries are "*Header File Only*" it can may sense to just copy the header(s) you need and store them in a directory that then gets added to the compiler search path for `#include`-d files.

> The major obstacle in this case are dependencies
> that some parts of Boost may have to other parts.

The task gets easier with the BCP utility:

- It is most easily used to simply extract header files and their dependencies.
- Besides that it can also extract source code and build tools.*

> For more information see: http://www.boost.org/doc/...
> .../libs/release/tools/bcp/doc/html/index.html

*: When applied to parts of Boost that aren't "*Header File Only*".

Configuring Boost

For the parts of boost that needs building (and for some header files that are systematically created) there are build options for fine-grained control of

- what is built and
- how it is built

More detailed instructions are provided in the "Getting Started" sections for Unix Variants and MicroSoft-Windows.

Online Compile Services

1. Coliru (Compile - Link - Run)
2. run code
3. Coding Ground
4. codepad
5. ideone
6. Webcompiler visual c++
7. Interactive Compile C++
8. Cloud9 IDE

Coliru - http://coliru.stacked-crooked.com/

Compilers / Libraries

- GCC: 6.2.0 (and many older versions too)
- Clang: 3.8.0 (and some older versions too)
- Boost: 1.61

User Interface

- Text area with syntax highlighting
- Command line to specify compile command*

Persistence / Collaboration

- Persistent links for sharing code

Special Features

*: (Static) standard input is possible via here-script.

run code - http://rextester.com/runcode

Compilers / Libraries

- GCC: 4.9.3
- Clang: 3.7.1
- Microsoft: 19.00.23506
- Boost: 1.54

User Interface

- Text area with syntax highlighting and choice of editor
- Prepared compile commands to select from drop-down menu

Persistence / Collaboration

- "Wall" to share code, Links for distribution

Special Features

Coding Ground - https://www.tutorialspoint.com/compile_cpp_online.php

Compilers / Libraries

- GCC: 5.3.1
- Clang: no
- Microsoft no
- Boost: no

User Interface

- Like a "Mini IDE"

Persistence / Collaboration

- Short URL can be created to distribute

Special Features

codepad - http://codepad.org/

Compilers / Libraries

- GCC: ?
- Clang: ?
- Microsoft: ?
- Boost: ?

User Interface

- Not intuitive of buggy (cannot modify code after first submit)

Persistence / Collaboration

- TBD

Special Features

ideone - http://ideone.com/

Compilers / Libraries

- GCC: 5.1
- Clang: ?
- Microsoft: ?
- Boost: 1.55

User Interface

- Single file in textarea window with syntax highlighting

Persistence / Collaboration

- TBD

Special Features

visual C++ - http://webcompiler.cloudapp.net/

Compilers / Libraries

- Microsoft: 19.10.24631.0
- Boost: ?

User Interface

- Single file in textarea window with syntax highlighting
- Compile flags can be added
- No interactive or prepared input, but command line arguments

Persistence / Collaboration

- TBD

Special Features

Automatically forwarded from prior *Visual C++ @ rise4fun from Microsoft*.

Compiler Explorer - http://gcc.godbolt.org/

Compilers / Libraries

- Rich set of compilers/versions - GCC based, Clang based, ICC, ...
- Targeting different hardware architectures - Intel, ARM, ...
- Command line options can be specified freely ...

User Interface

- One window shows source, the other one assembler output.
- Correspondence between C++ and assembler source may optionaly be elucidated through colorizing.

Persistence / Collaboration

- Local Browser Store

Special Features

Does **not** compile to run an executable but to show assembler code!

Cloud9 IDE - https://c9.io/

Compilers / Libraries

- TBD

User Interface

- TBD

Persistence / Collaboration

- TBD

Special Features

Not just an online compiler but a complete cloud-based IDE.