



# Guide to the Modern C++ Ecosystem

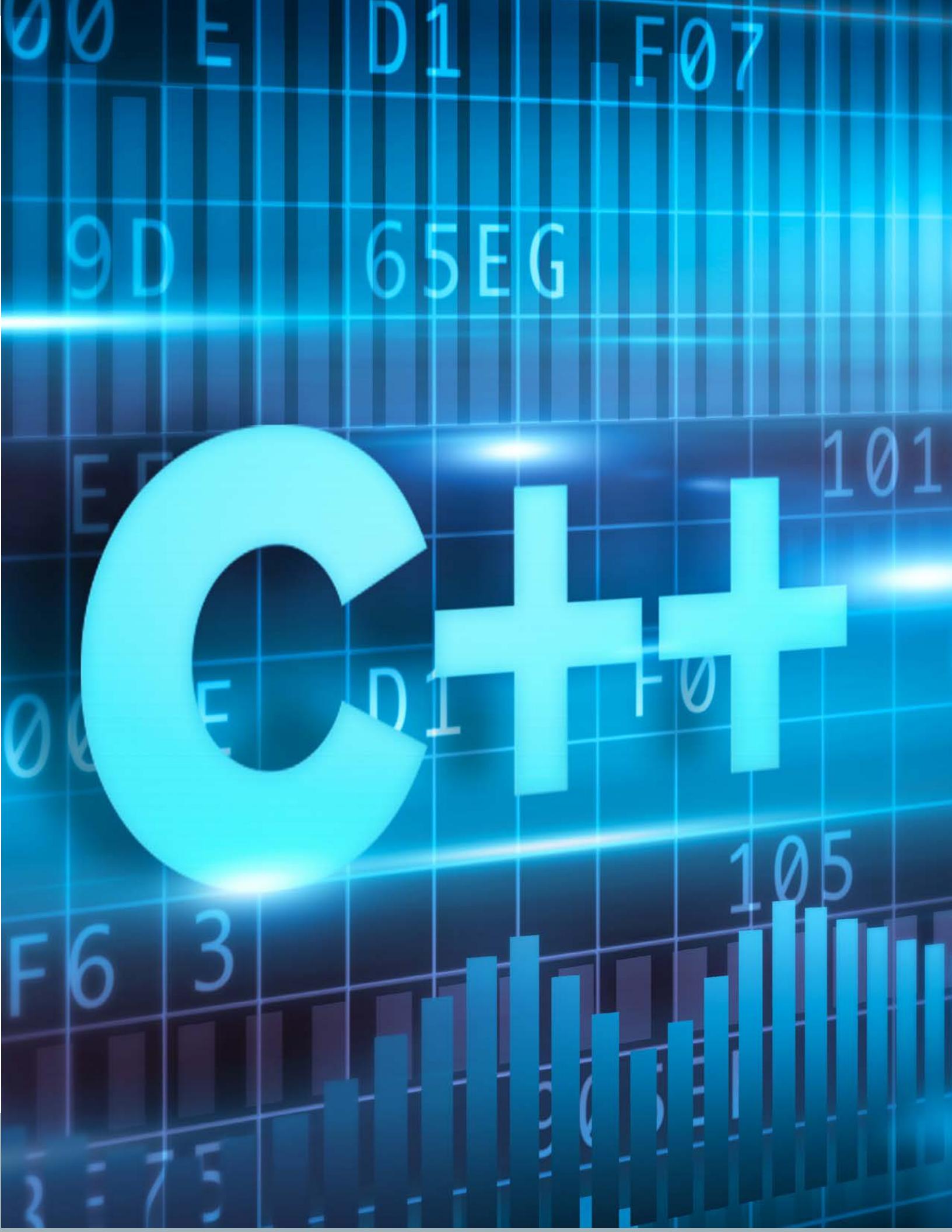
# MODERN C++

The C++ computer programming language has become one of the most widely used modern programming languages. Software built with C++ is known for its performance and efficiency. C++ has been used to build numerous vastly popular core libraries, applications such as Microsoft Office, game engines such as Unreal, software tools like Adobe Photoshop, compilers like Clang, databases like MySQL, and even operating systems such as Windows across a wide variety of platforms as it continues to evolve and grow.

Modern C++ is generally defined as C++ code that utilizes language features in C++11, C++14, and C++17 based compilers. These are language standards named after the year they were defined (2011, 2014 and 2017 respectively) and include a number of significant additions and enhancements to the original core language for powerful, highly performant, and bug free code. Modern C++ has high level features that support object oriented programming, functional programming, generic programming, and low level memory manipulation features.

Big names in the computer industry such as Microsoft, Intel, the Free Software Foundation, and others have their own modern C++ compilers. Companies such as Microsoft, The QT Company, JetBrains, and Embarcadero provide integrated development environments for writing code in modern C++. Popular libraries are available for C++ across a wide range of computer disciplines including Artificial Intelligence, Machine Learning, Robotics, Math, Scientific Computing, Audio Processing, and Image Processing.

In this guide we are going to cover a number of these compilers, build tools, IDEs, libraries, frameworks, coding assistants, and much more that can support and enhance your development with modern C++. Let's get started!



# COMPILERS, BUILD TOOLS, & PACKAGE MANAGERS

There are a number of popular compilers that support modern C++ including gcc/g++, msvc (Microsoft Visual C++), and Clang. Each compiler has varying support for each of the major operating systems with the open source gcc/g++ originating in the late 1980s, Microsoft's Visual C++ in the early 1990s, and Clang in the late 2000s. All three compilers support modern C++ up to at least C++17 but the source code licenses for each of them vary greatly.



## GCC

gcc is a general-use compiler developed and maintained and regularly updated by the GCC Steering committee as part of the GNU Project. gcc describes a large growing family of compilers targeting many hardware platforms and several languages. While it mainly targets Unix-like platforms, Windows support is provided through the Cygwin or MinGW runtime libraries. gcc compiles modern C++ code up to C++17 with experimental support for some C++20 features. It also compiles with a variety of language extensions that build upon C++ standards. It is free and open source (GPL3) with the gcc Runtime Library Exception. gcc has support from build tools such as CMake and Ninja and many IDEs such as CLion, Qt Creator, and Visual Studio Code.

<https://gcc.gnu.org/>

# MICROSOFT VISUAL C++

Microsoft Visual C++ (MSVC) is Microsoft's compiler for their custom implementation of the C++ standard, known as Visual C++. It is regularly updated, and like gcc and Clang, supports modern C++ standards up to C++17 with experimental support for some C++20 features. MSVC is the main method for building C++ applications in Microsoft's own Visual Studio. It generally targets a number of architectures on Windows, Android, iOS, and Linux. Support for build tools and IDEs are limited but growing. CMake extensions are available in Visual Studio 2019. MSVC can be used with Visual Studio Code, with limited support from CLion and Qt Creator with additional extensions. MSVC is proprietary and available under a commercial license.

[https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_C%2B%2B](https://en.wikipedia.org/wiki/Microsoft_Visual_C%2B%2B)

# CLANG

Clang describes a large family of compilers for the C family of languages maintained and regularly developed as part of the LLVM project. Although it targets many popular architectures, it generally targets fewer platforms than gcc. The LLVM project defines Clang through key design principles - strict adherence to C++ standards (although support for gcc extensions is offered), modular design, and minimal modification to the source code's structure during compilation, to name a few. Like gcc, Clang compiles modern C++ code with support for the C++17 standard with experimental C++20 support. It is available under an open source (Apache License Version 2.0) license. Clang also has widespread support from build tools such as CMake and Ninja and IDEs such as CLion, Qt Creator, Xcode, and others.

<https://clang.llvm.org/>

# COMPILER EXPLORER

Compiler Explorer is a web based tool that allows you to select from a wide variety of C++ compilers and different versions of the same compiler to test out your code. This allows developers to compare the generated code for specific C++ constructs among compilers, and test for correct behaviour. Clang, gcc, and msvc are all there but also lesser-known compilers such as DJGPP, ELLCC, Intel C++, and others.

<https://godbolt.org/>

# CMAKE

CMake is a cross platform tool for managing your build process. Building, especially large apps and with dependent libraries, can be a very complex process especially when you support multiple compilers; CMake abstracts this. You can define complex build processes in one common language and convert them to native build directives for any number of supported compilers, IDEs, and build tools, including Ninja (below.) There are versions of CMake available for Windows, macOS, and Linux.

<https://cmake.org/>

## NINJA

The Ninja build system is used for the actual process of building apps and is similar to Make (a traditional but now less used utility.) It focuses on running as fast as possible by parallelizing builds. It is commonly used paired with CMake, which supports creating build files for the Ninja build system. The feature set of Ninja is intentionally kept minimal because the focus is on speed.

<https://ninja-build.org/>

## MICROSOFT BUILD ENGINE (MSBUILD)

MSBuild is a command line based built platform available from Microsoft under an open source (MIT) license. It can be used to automate the process of compiling and deploying projects. It is available stand alone, packaged with Visual Studio, or from Github. The structure and function of MSBuild files is very similar to Make. MSBuild has an XML based file format and mainly has support for Windows but also macOS and Linux. IDEs such as CLion and C++Builder can integrate with MSBuild as well.

<https://docs.microsoft.com/en-us/visualstudio/msbuild/msbuild>

## CONAN, VCPKG, BUCKAROO

Package managers such as Conan, vcpkg, and Buckaroo have been gaining popularity in the C++ community. A package manager is a tool to install libraries or components. Conan is a decentralized open source (MIT) package manager that supports multiple platforms and all build systems (such as CMake and MSBuild). Conan supports binaries with a goal of automating dependency management to save time in development and continuous integration. Microsoft's vcpkg is open source under an MIT license and supports Windows, macOS, and Linux (and even integrates with CMake). Vcpkr requires Visual Studio 2015 or newer so it is somewhat limited in scope. Buckaroo is a lesser known open source package manager that can pull dependencies from GitHub, BitBucket, GitLab, and others. Buckaroo offers integrations for a number of IDEs including CLion, Visual Studio Code, XCode, and others.

<https://conan.io/>

<https://github.com/microsoft/vcpkg>

<https://buckaroo.pm/>

# INTEGRATED DEVELOPMENT ENVIRONMENTS

A host of editors and integrated development environments (IDEs) can be used for developing with modern C++. Text editors are typically lightweight, but are less featureful than a full IDE and so are used only for the process of writing code, not debugging or testing it. Full development requires other tools, and an IDE contains those and integrates into cohesive integrated development environment. Any number of text editors like Sublime Text, Atom, Visual Studio Code, vi/vim, and emacs can be used for writing C++ code. However, some IDEs are specifically designed with modern C++ in mind like CLion, Qt Creator, and C++Builder, while IDEs like Xcode and Visual Studio also support other languages.



## SUBLIME TEXT, ATOM, AND VISUAL STUDIO CODE

Sublime Text is a commercial text editor with extended support for modern C++ available via plugins. Atom is an open source (MIT license) text editor that supports modern C++ via packages with integrations available for debugging and compiling. Visual Studio Code is a popular open source (MIT license) source-code editor from Microsoft. A wide variety of extensions are available that bring features such as debugging and code completion for modern C++ to Visual Studio Code. Sublime Text, Atom, and Visual Studio Code are all available for Windows, macOS, and Linux.

<https://www.sublimetext.com/>

<https://atom.io/>

<https://code.visualstudio.com/>

## VI/VIM & EMACS

Vi/Vim and Emacs are free command line based text editors that are mainly used on Linux but are also available for macOS and Windows. Modern C++ support can be added to Vi/Vim through the use of scripts while modern C++ support can be added to Emacs through the use of modules.

<https://www.vim.org/>

<https://www.gnu.org/software/emacs/>

## CLION

CLion is a commercial IDE from JetBrains that supports modern C++. It can be used with build tools like CMake and Gradle, integrates with the GDB and LLDB debuggers, can be used with version control systems like Git, test libraries like Boost.Test, and various documentation tools. It has features such as code generation, refactoring, on the fly code analysis, symbol navigation, and more.

<https://www.jetbrains.com/clion/>

## QT CREATOR

Qt Creator is a commercial IDE from The Qt Company which supports Windows, macOS, and Linux. Qt Creator has features such as a UI designer, syntax highlighting, auto completion, and integration with a number of different modern C++ compilers like gcc and CLang. Qt Creator closely integrates with the Qt library for rapidly building cross platform applications. Additionally, it integrates with standard version control systems like Git, debuggers like GDB and LLDB, build systems like CMake, and can deploy cross platform to iOS and Android devices.

<https://www.qt.io/>

## C++BUILDER

C++Builder is a commercial IDE from Embarcadero Technologies which runs on Windows and supports modern C++. It features the award winning Visual Component Library (VCL) for Windows development and FireMonkey (FMX) for cross platform development for Windows, iOS and Android. The C++Builder compiler features an enhanced version of Clang, an integrated debugger, visual UI designer, database library, comprehensive RTL, and standard features like syntax highlighting, code completion, and refactoring. C++Builder has integrations for CMake, can be used with Ninja, and also MSBuild.

<https://www.embarcadero.com/products/cbuilder>

# VISUAL STUDIO

Visual C++ is a commercial Visual Studio IDE from Microsoft. Visual Studio integrates building, debugging, and testing within the IDE. It provides the Microsoft Foundation Class (MFC) library which gives access to the Win32 APIs. Visual Studio features a visual UI designer for certain platforms, comes with MSBuild, supports CMake, and provides standard features such as code completion, refactoring, and syntax highlighting. Additionally, Visual Studio supports a number of other programming languages and the C++ side of it is focused on Windows, with other platforms slowly being added.

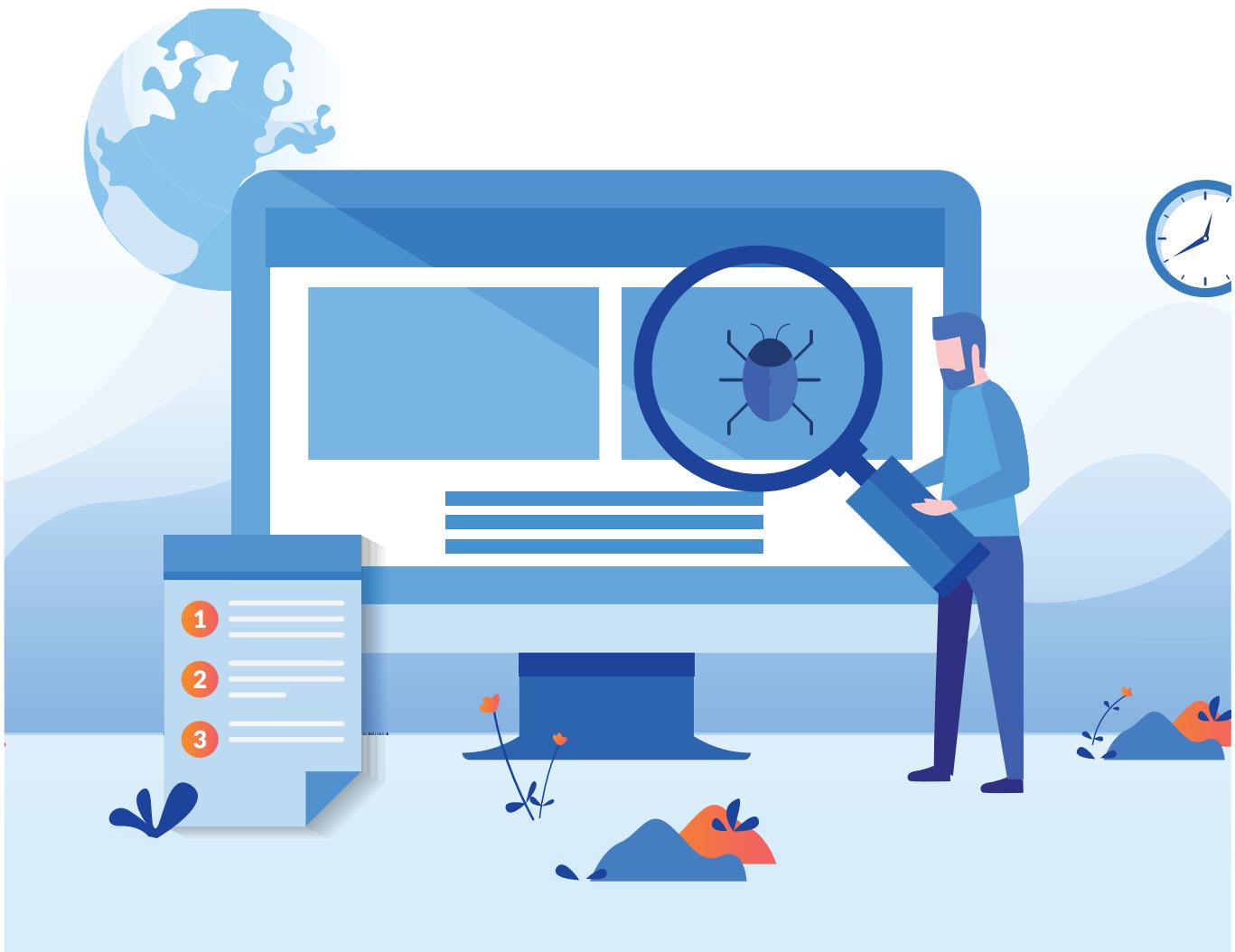
<https://visualstudio.microsoft.com/>

# XCODE

Xcode is a multi-language IDE from Apple available only on macOS that supports modern C++. Xcode is proprietary but available for free from Apple. Xcode has an integrated debugger, supports version control systems like Git, features a Clang compiler, and utilizes libc++ as its standard library. It supports standard features such as syntax highlighting, code completion, and refactoring. Additionally, Xcode supports external build systems like CMake and utilizes the LLDB debugger.

<https://developer.apple.com/xcode/>

# DEBUGGING & TESTING



## GDB

GDB is a portable command line based debugging platform that supports modern C++ and is available under an open source license (GPL). A number of editors and IDEs like Visual Studio, Qt Creator, and CLion support integration with GDB. It can also be used to debug applications remotely where GDB is running on one device and the application being debugged is running on another device. It supports a number of platforms including Windows, macOS, and Linux.

<https://www.gnu.org/software/gdb/>

## LLDB

LLDB is an open source debugging interface that supports modern C++ and integrates with the Clang compiler. It has a number of optional performance enhancing features such as JIT but also supports debugging memory, multiple threads, and machine code analysis. It is built in C++. LLDB is the default debugger for Xcode and can be used with Visual Studio Code, CLion, and Qt Creator. It supports a number of platforms including Windows, macOS, and Linux.

<https://lldb.llvm.org/>

## CATCH/CATCH2

Catch2 is a cross platform open source (BSL-1.0) testing framework for modern C++. It is very lightweight because only a header file needs to be included. Unit tests can be tagged and run in groups. It supports both test-driven development and behavior-driven development. Catch2 also easily integrates with CLion.

<https://github.com/catchorg/Catch2>

## BOOST.TEST

Boost.Test is a feature rich open source (BSL-1.0) testing framework that utilizes modern C++ standards. It can be used to quickly detect errors, failures, and time outs through customizable logging and real time monitoring. Tests can be grouped into suites and supports both small scale testing and large scale testing.

<https://github.com/boostorg/test>

## GOOGLE TEST

Google Test is Google's C++ testing and mocking framework which is available under an open source (BSD) license. Googletest can be used on a broad range of platforms including Linux, macOS, Windows, and others. It contains a unit testing framework, assertions, death tests, detects failures, handles parameterized tests, and creates XML test reports.

<https://github.com/google/googletest>

# STANDARD LIBRARIES & FRAMEWORKS



## STANDARD TEMPLATE LIBRARY

The Standard Template Library or STL is part of the C++ standard provides portable common data structures and algorithms. Data structures such as queues, lists, and iterators are provided in very efficient implementations. Algorithms for searching, sorting, file system access and more are also available and very efficient. Modern C++ compilers are configured to minimize performance penalties which may arise from using the STL extensively. There are a number of different implementations of the STL including libstdc++, libc++, Dinkumware, Microsoft STL, and others.

[https://en.wikipedia.org/wiki/Standard\\_Template\\_Library](https://en.wikipedia.org/wiki/Standard_Template_Library)

## BOOST

Boost is an open source (Boost Software License) library for modern C++ which provides a wide variety of functionality extending the STL. It is actually a collection of a large number of other libraries which are peer-reviewed before being added to Boost. Everything from smart pointers, to file system abstractions, to template metaprogramming, many maths and engineering functions, and beyond. A lot of the Boost libraries are header based and Boost enjoys a wide compatibility with tools, IDEs, and compilers across the breadth of the modern C++ ecosystem.

<https://www.boost.org/>

## EIGEN

Eigen is an open source (MPL2) library that can be used with modern C++ that focuses on linear algebra. It features template headers for things like matrices, vectors, geometrical transformations, and more. Eigen was designed to be versatile, fast, reliable, and elegant. A number of SIMD instruction sets are supported including SSE and ARM NEON. It has good support for gcc, Visual C++, Clang, and a number of other compilers.

<http://eigen.tuxfamily.org/>

## DLIB

Dlib is another broad well known library that is open source under the Boost Software License. It is designed for academic and industry uses including functionality relating to machine learning, scientific computation, image processing, computer vision, data compression, and networking. Dlib has lots of examples, good documentation, wide unit testing coverage, no outside dependencies, and is very portable. It is regularly tested across Windows, macOS, and Linux.

<http://dlib.net/>

## QT

The Qt library for modern C++ is an extensive cross platform library for building applications that support a wide range of platforms such as Windows, macOS, Linux, iOS, and Android. It has a wide variety of modules for graphics, data storage, multimedia, networking, location, inputs, web content, and language bindings. It pairs with the Qt Creator IDE from The Qt Company. It is available under a number of licenses including commercial and open source licenses (GPL2, GPL3, and LGPL3).

<https://www.qt.io/>

# USER INTERFACE (UI) FRAMEWORKS



## C++/WINRT

C++/WinRT is modern C++ header library available under the open source MIT license from Microsoft for the Windows Runtime (WinRT). C++/WinRT is part of the Microsoft Windows SDK and mainly pairs with Visual Studio and the Visual C++ compiler. One goal of C++/WinRT is to provide a modern C++ language projection of the Windows RunTime and its COM APIs. C++/WinRT uses a lot of modern C++ features to increase performance and development productivity.

<https://github.com/microsoft/cppwinrt>

## GTK

GTK is an open source (LGPL) cross platform UI library with interfaces for modern C++ available through GTKMM (also available under LGPL). GTK is mainly used in the Linux world but is also available on Windows, macOS, and other platforms. GTKMM offers full UTF8 internationalisation, uses inheritance for the UI objects, makes full use of C++ namespaces, uses the C++ Standard Library for strings, containers, and iterators, and offers type safe standard C++ signal handlers.

<https://www.gtk.org/> - <https://www.gtkmm.org/>

## SCITER

Sciter is a commercial multiplatform embeddable graphics engine library for desktop UI development with HTML and CSS available to modern C++. Sciter supports Windows, macOS, and Linux (via GTK3). It is used by a number of popular applications and has a small binary footprint. Sciter provides rendering accelerated via the GPU as well.

<https://sciter.com/>

## QT

The Qt library for modern C++ is a cross platform widget library for building UI based applications that support a wide range of platforms such as Windows, macOS, Linux, iOS, and Android. It is available under a number of licenses including commercial and open source licenses (GPL2, GPL3, and LGPL3) and pairs with the Qt Creator development tool by The Qt Company. Qt features a declarative UI language (QML) and offers rendering via a software rasterizer, OpenGL ES, or OpenGL.

<https://www.qt.io/>

## JUCE

JUCE is an open source (GPL) and commercial application framework for modern C++. It supports the Windows, macOS, and Linux platforms and generally officially supports the gcc, clang, and Microsoft Visual C++ compilers. JUCE has a range of functions that cover UI, XML, JSON, networking, threading, audio, and graphics. JUCE has a large set of audio functions which is one area where it excels.

<https://juce.com/>

## VISUAL COMPONENT LIBRARY

Visual Component Library (VCL) is an award winning object oriented framework commercially available for use in modern C++ within Embarcadero's C++Builder IDE. The VCL wraps a good portion of the Microsoft Windows Win32 APIs and has a large number of visual and nonvisual controls and components for use on Windows, and is notable for providing a platform-native experience. The VCL library also extends existing Windows controls providing additional functionality and offers data binding of controls. Windows applications can be visual designed when paired with the IDE for rapid application development.

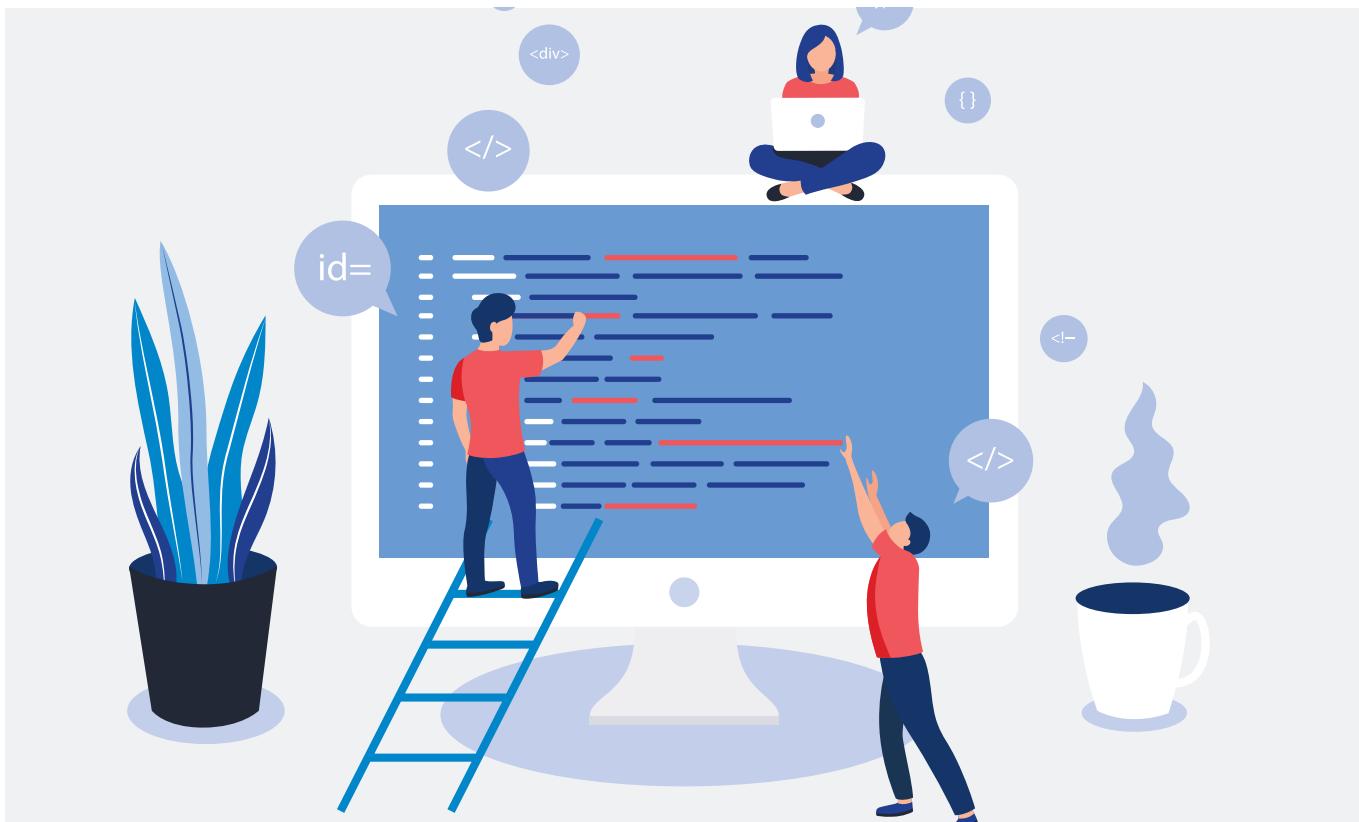
[http://docwiki.embarcadero.com/RADStudio/Rio/en/VCL\\_Overview](http://docwiki.embarcadero.com/RADStudio/Rio/en/VCL_Overview)

## FIREMONKEY

FireMonkey (FMX) is a cross platform framework commercially available with Embarcadero's C++Builder IDE and compiler which supports modern C++. FMX is an abstraction layer for each platform's UI elements which are drawn via GDI+, Direct2D, OpenGL, or OpenGL ES depending on the platform. FireMonkey supports working with graphics, animations, audio, and direct API access to the underlying platform specific APIs. It can be used on Windows, iOS, and Android with C++.

<http://docwiki.embarcadero.com/RADStudio/Rio/en/FireMonkey>

# CODING ASSISTANCE



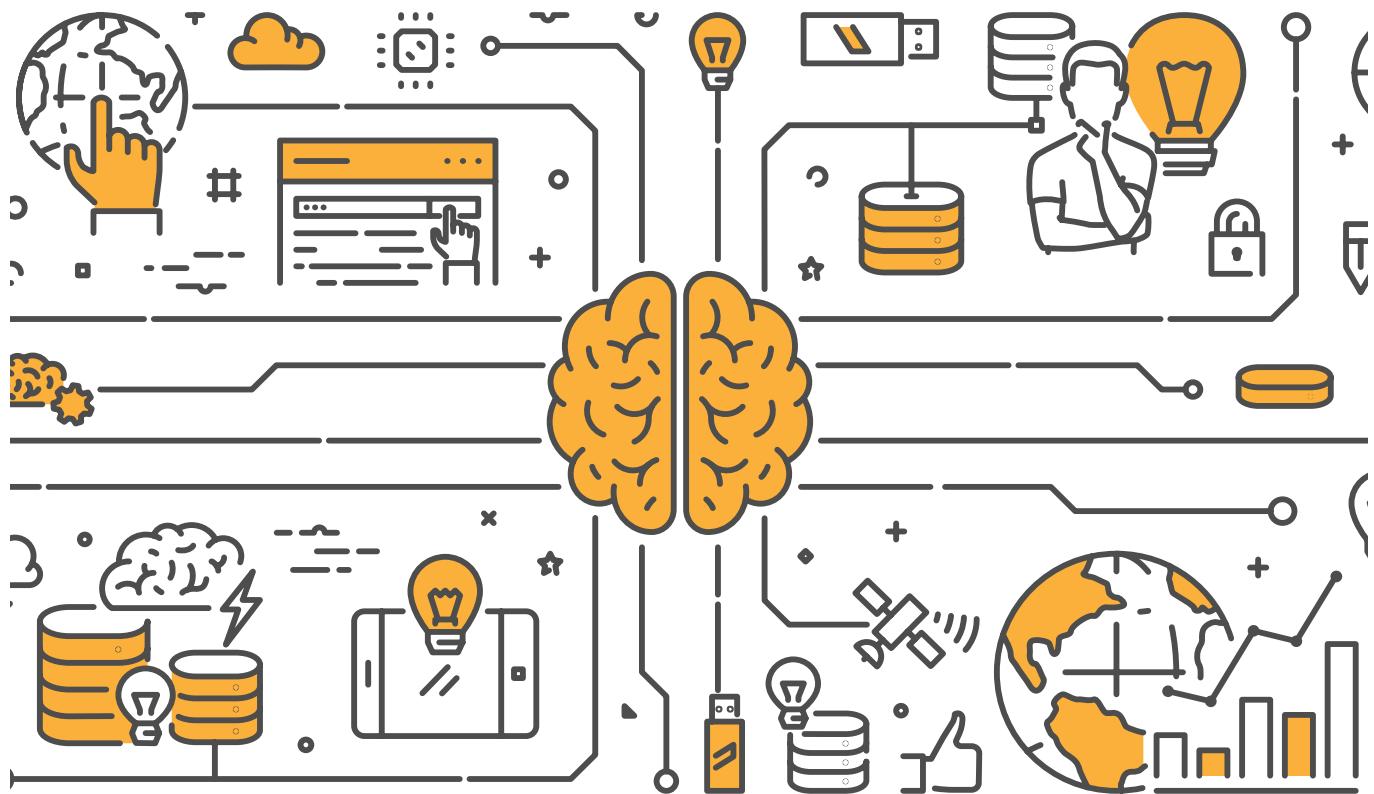
## VISUAL ASSIST BY WHOLE TOMATO

Visual Assist is a commercial code assisting extension for Visual Studio that is designed to enhance developer productivity with modern C++. It has unique features including excellent code completion and refactoring that are more powerful than the built-in Visual Studio tools, navigation features, custom tags for navigation, and extended syntax highlighting. It can be used to reduce the complexity of code, improve on its readability, make it extensible, and is popular among those dealing with large codebases like operating systems and games.  
<https://www.wholtomato.com/>

## CLANG-TIDY

Clang-tidy is a linter tool for modern C++ available under the open source Apache 2.0 license. It was built to provide a framework that can be extended for finding and fixing common errors when writing C++ code. It is modular and helps with fixing style violations and misuse of interfaces plus it uses static analysis to help locate bugs.  
<https://clang.llvm.org/extra/clang-tidy/>

# COMPUTATION



## ARTIFICIAL INTELLIGENCE (TENSORFLOW, TINY-DNN, FRUGALLY-DEEP)

The Tensorflow C++ API provides all the tools you need to design and train deep learning models. It leverages the expressiveness of modern C++ standards, the speed of optimized C++ implementations, and behind-the-scenes CUDA support for GPU-aided computation. Modern C++ libraries also offers more lightweight artificial intelligence tools. Tiny-DNN uses optimized algorithms to bring Deep Learning technology to devices with limited computing capability. The Frugally-Deep library lets you leverage the more simple, accessible Keras Deep Learning API combined with a minimal-dependency C++ binary for fast prototyping.

<https://www.tensorflow.org/guide/extend/cc>

<https://github.com/tiny-dnn/tiny-dnn>

<https://github.com/Dobiasd/frugally-deep>

## MACHINE LEARNING (DLIB, XGBOOST, MLPACK)

The popular Dlib library can be used for creating complex software as it provides various machine learning algorithms. The portable library XGBoost can be used to solve complex data science problems accurately and quickly. The tools provided in the mlpack library can be used to implement extensible versions of modern machine learning algorithms quickly.

<http://dlib.net/>

<https://github.com/dmlc/xgboost>

<https://www.mlpack.org/>

## ROBOTICS (ROBOTICS LIBRARY, MRPT)

Modern C++ can be used for robotics research and applications. The Robotics Library is a robust modern C++ package providing tools for efficient physics calculations, abstractions for hardware interaction, and 3D visualizations. MRPT offers modern C++ libraries with built-in hardware support for a range of sensors and devices, along with a wealth of pre-made, ready-to-use robotics programs - tools computer vision, motion tracking with Microsoft Kinect, and more.

<https://www.mrpt.org/>

<https://www.roboticslibrary.lorg/>

## MATH (ARMADILLO, CERES SOLVER, XTENSOR)

The speed of modern C++ can be used to build production-ready engineering applications working with many variables and constraints. Armadillo provides fast implementations of linear algebra operations with modern C++. Ceres Solver offers tools for many kinds of multivariate optimization problems in modern C++. XTensor is a modern C++ library to transform and summarize multi-dimensional data with optimized execution and memory usage.

<http://arma.sourceforge.net/>

<http://ceres-solver.org/>

<https://github.com/QuantStack/xtensor>

## SCIENTIFIC COMPUTING (TRILINOS, TILEDDB, DEAL.II)

In addition to engineering applications, modern C++ libraries bring efficiency to computation-heavy research projects. Trilinos provides a suite of modern C++ libraries to meet the requirements of different research settings, such as large-scale parallel computation or manipulating complex geometry and mesh data. TileDB provides modern C++ utilities for working with massive matrix data, with support for PrestoDB for Big Data SQL and distributed storage through HDFS or Amazon Web services. Deal.II provides an efficient modern C++ framework for using the finite element method to solve complex problems in engineering areas such as fluid and heat dynamics.

<https://github.com/trilinos/Trilinos>

<https://tiledb.io/>

<https://www.dealii.org/>

## AUDIO PROCESSING (WAV2LETTER, KFR)

Modern C++ makes available fast, complex audio processing. Modern C++ developers can integrate fast, state-of-the-art speech recognition through wav2letter++, an open-source C++ library developed by Facebook's research team. KFR is a robust, open-source C++ library for digital signal processing applications.

<https://github.com/facebookresearch/wav2letter>

<https://www.kfrlib.com/>

# DATABASE



## SQLAPI++

SQLAPI++ is a commercial database access library for C++. It supports connecting to Oracle, SQL Server, DB2, Sybase, Informix, InterBase, SQLBase, MySQL, PostgreSQL, SQLite, SQL Anywhere, and ODBC. A number of platforms are listed that it supports including Windows, macOS, and Linux. SQLAPI++ provides a low level interface for database specific features, and natively calls DBMS APIs. It lists support for both the gcc and Microsoft Visual C++ compilers in addition to a couple others.

<http://www.sqlapi.com/>

## POCO C++

POrtable COmponents (POCO) C++ is a collection of cross platform libraries available under an open source (Boost Software License) license for building internet and network based apps using modern C++. It has functionality for threading, logging, and encryption but it also has a database library for accessing SQLite, MySQL/MariaDB, PostgreSQL and SQL Server (via ODBC), MongoDB, and Redis. Windows, Linux, and macOS are supported in addition to a few other platforms. POCO C++ can be built with CMake and Make, has solutions for Visual Studio, and is available via Github, Conan, and vcpkg.

<https://pocoproject.org/>

## SOCI

SOCI (Simple Oracle Call Interface) is an open source (Boost Software License) database access library which uses some modern C++ language features and the Boost library. It supports DB2, Firebird, MySQL, ODBC, Oracle, PostgreSQL, SQLite3. Three major compilers are supported by the library including gcc, Microsoft Visual C++, and Clang. Linux, macOS, and Windows platforms all look to be compile targets for the library.

<https://github.com/SOCI/soci>

## FIREDAC

FireDAC is a native universal data access library that supports modern C++ for building applications on Windows, Android, and iOS that connect to enterprise databases. It gives you native high speed direct access to InterBase, SQLite, MySQL, SQL Server, Oracle, PostgreSQL, DB2, SQL Anywhere, Advantage DB, Firebird, Access, Informix, MongoDB, ODBC, and others. It has a common API for accessing different database backends that abstracts and simplifies data access while utilizing unique database specific features. FireDAC is a commercial library available with Embarcadero's C++Builder.

<https://www.embarcadero.com/products/rad-studio/firedac>

# EXCITED ABOUT MODERN C++? SAVE TIME WITH: EMBARCADERO C++BUILDER...

Embarcadero C++Builder delivers some of the best modern C++ features and tooling all wrapped up into one integrated package so you can save time and money.

- **BEAUTIFUL USER INTERFACES** - Design beautiful intuitive user interfaces with award winning and cross platform frameworks on **Windows, iOS, and Android**. Provide a platform-native experience to your users while designing the interfaces only once, cutting development time in half or more.
- **WRITE BETTER CODE** - Write better code faster using modern OOP practices, robust frameworks, and the feature rich C++Builder IDE. It has a **modern C++17 Clang-enhanced compiler**, tool chain, and the **Dinkumware standard template library plus advanced code completion**.
- **FIND BUGS FASTER** - Find bugs faster with the integrated cross-platform native debugging tools. Debug remotely on **Windows, iOS, and Android** devices.
- **BEST CODING PRACTICES** - Enforce best coding practices, reduce duplication of effort, and become a coding rock star with faster development cycles using C++Builder.
- **CREATE BETTER PRODUCTS FASTER** - Integrate with continuous build configurations quickly with **MSBuild, CMake, and Ninja** project support either as a lone developer or part of a team.
- **HUNDREDS OF COMPONENTS** - Utilize popular libraries such as **Boost, Eigen, and more**, hundreds of built in components from creating user interfaces to database connectivity for building connected cross platform applications faster, and leverage a broad range of community tools and libraries.
- **DATA ACCESS** - Get native high-speed direct access to InterBase, SQLite, MySQL, SQL Server, Oracle, PostgreSQL, DB2, SQL Anywhere, Advantage DB, Firebird, Access, Informix, MongoDB, and more.

## WHAT ARE YOU WAITING FOR?

**START FOR FREE**

# VISUAL ASSIST

## BY WHOLE TOMATO

An extension for Microsoft Visual Studio that provides productivity features to turn a good IDE into a great IDE. Add flexibility and efficiency to your Microsoft Visual Studio instance for C/C++ and C# development, making your dev teams more efficient.

- **EASY NAVIGATION** - Move about your code with newfound ease—to any file, method, symbol, or reference in your projects and solutions. Get anywhere from anywhere. The navigation features of Visual Assist, like the other types of features, are entirely new to the IDE, improve built-in functionality, or in a few instances, apply only to older IDEs.
- **REFACTORING AT ITS FINEST** - Reduce the complexity of your code, improve its readability, and make it extensible without changing its external behavior. Refactor legacy code, the code you inherit, or the first version of your own work.
- **ACCELERATED CODE GENERATION** - Generate code faster than you think possible, even if you use just a handful of the features in Visual Assist. As you become a proficient user, write and update code even faster.
- **CODING ASSISTANCE FOR PEACE OF MIND** - Write code faster with assistance that doesn't get in your way, especially when you're on a roll. Visual Assist suggests completions only if they save you serious typing, and quietly fixes mistakes in your wake.
- **KNOW YOUR CODE** - Make sense of complex code, or code you haven't touched in a long time, with an assortment of tool windows, browsers, and fields that provide information where and when you need it.
- **CODE ERROR REMEDIATION** - Watch Visual Assist correct mistakes as you make them, especially errors in symbol case and pointer notation. Type entirely in lower case, and watch Visual Assist correct the case of all of your symbols. Save the shift key for your definitions.
- **AND MUCH MORE...**

## WHAT ARE YOU WAITING FOR?

**START FOR FREE**



An Idera, Inc. Company

CX DX RX