

Introducing

RAD Studio

12.1 Athens

RAD Studio 12 Athens was a *Mega Release*

1. C++
 - Visual Assist
 - New Clang Tech Preview
2. Installer & IDE
 - Multidevice Icons
 - DelphiLSP
 - VCL Designers, IDE Misc, ToolsAPI
3. FireMonkey & Skia
 - Skia UI Controls and rendering and graphic formats and way more (and VCL as well)
 - TMemo and TEdit
4. VCL
 - MDI and Tab based UIs
 - Fonts and Screen, Components improvements
4. Data
 - JSON Data Binding
 - FireDAC Secure Coding
 - Sqids, RAD Server, HTTP, REST
5. Delphi
 - String literals improvements
 - Platforms (Windows, Android)
 - Circular Uses Statements, Floating Point
6. Quality
 - *How many bug fixes this release?!*

The focus of RAD Studio 12.1 is on improving the quality of the many new features added in RAD Studio 12 Athens and on completing the Clang upgrade (moving it from preview to full feature)

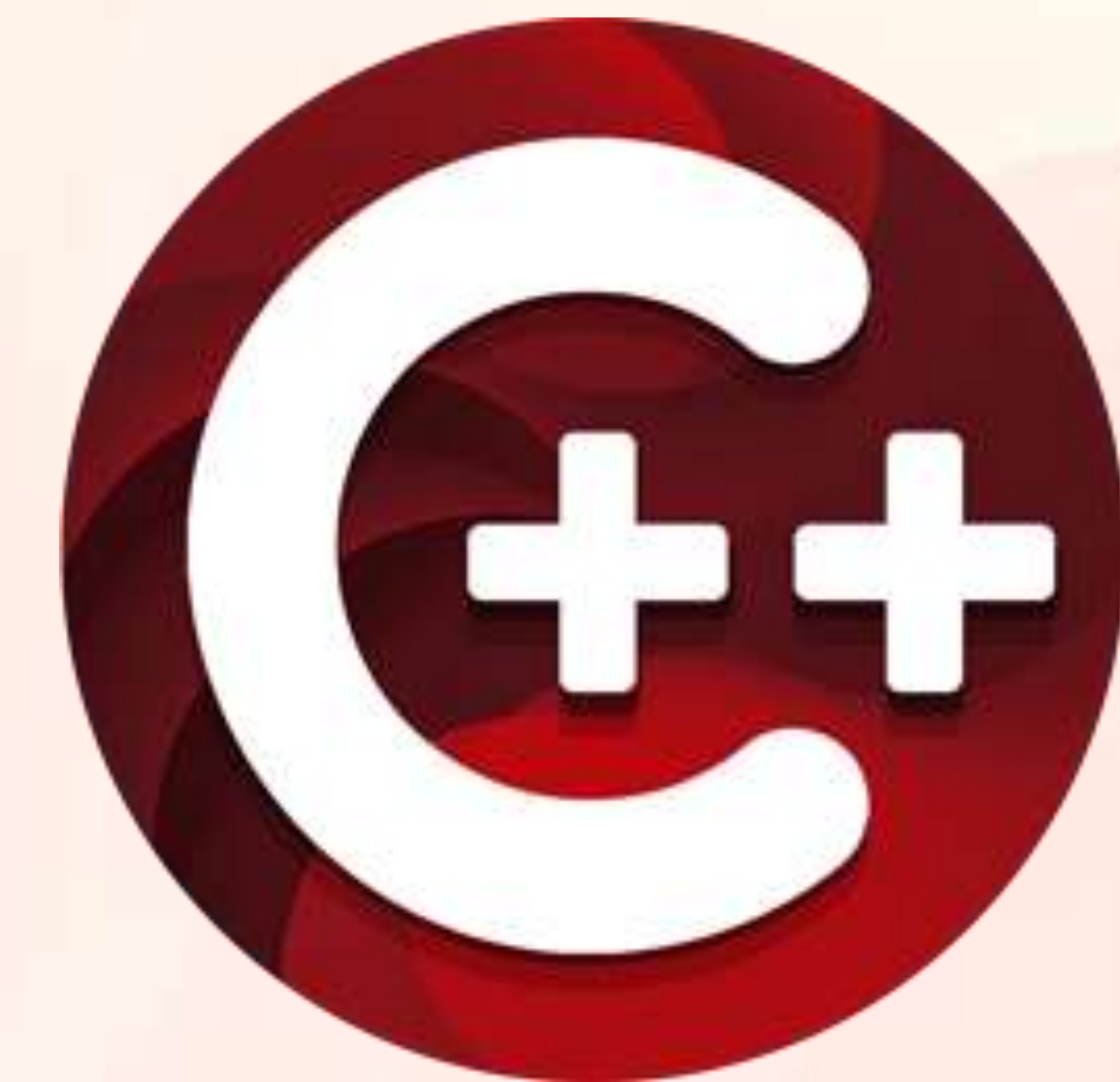
RAD 12.1 Summary: Focus on Quality

- **Some substantial features**
 - Editor split views and other IDE enhancements
 - The delivery of the new C++Builder Win64 Clang compiler
 - Android API level 34 support
- **Quality highlights include**
 - C++ Builder VA integration
 - VCL and extensive FireMonkey quality
 - RTL, database, etc
 - Includes the 12.0 Patch 1 Fixes (released in January)



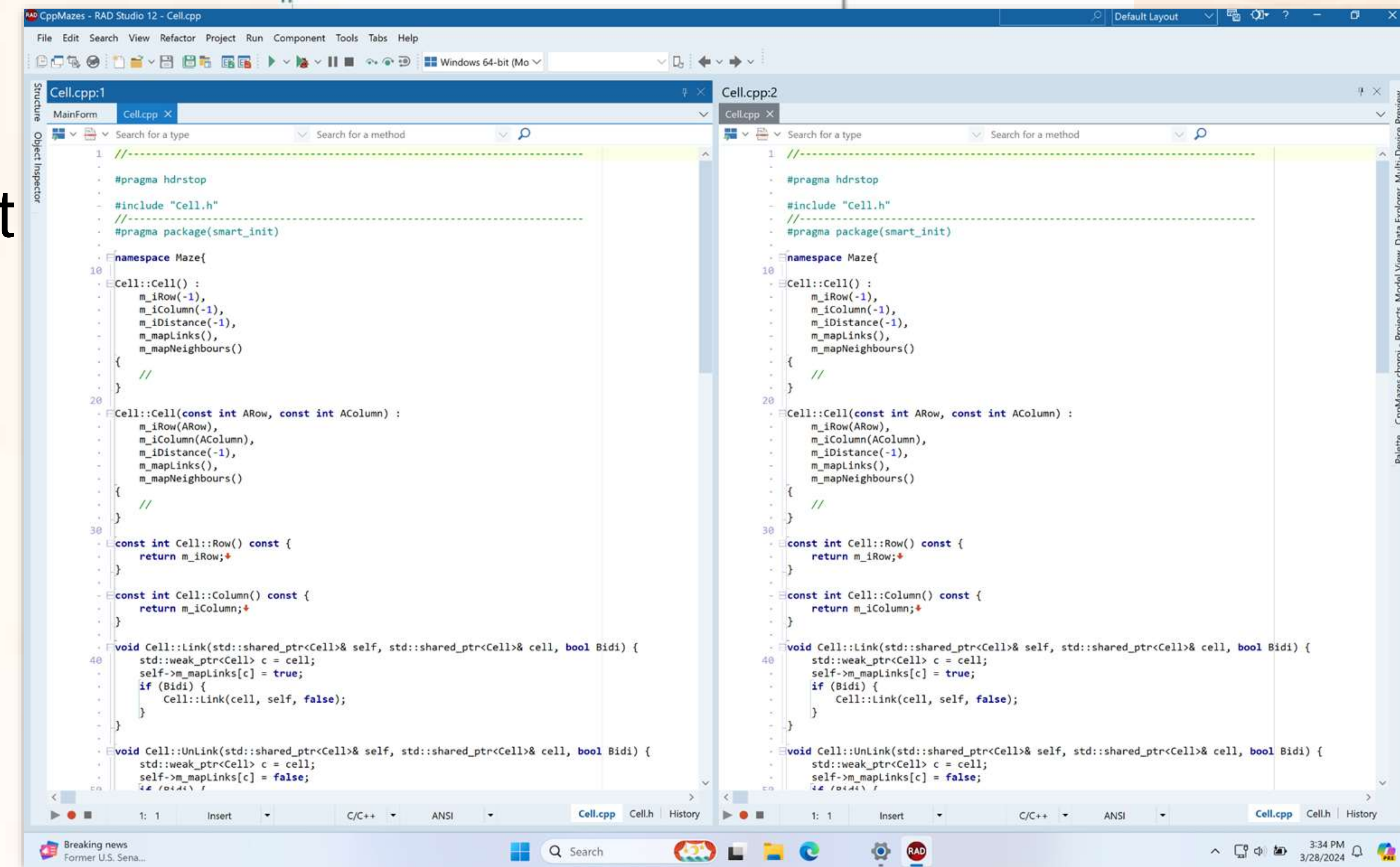
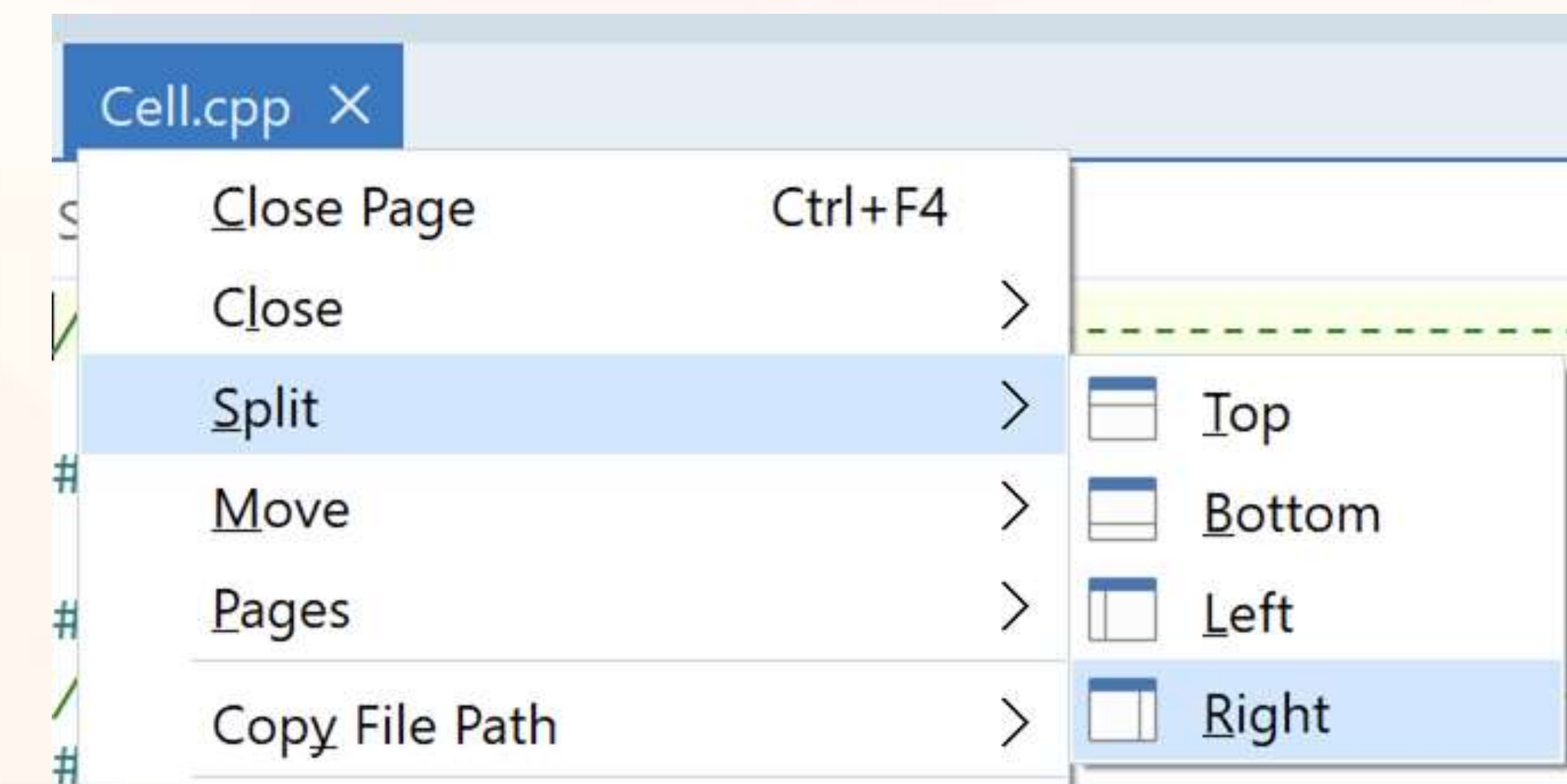
IDE Improvements

Demo: Editor split views



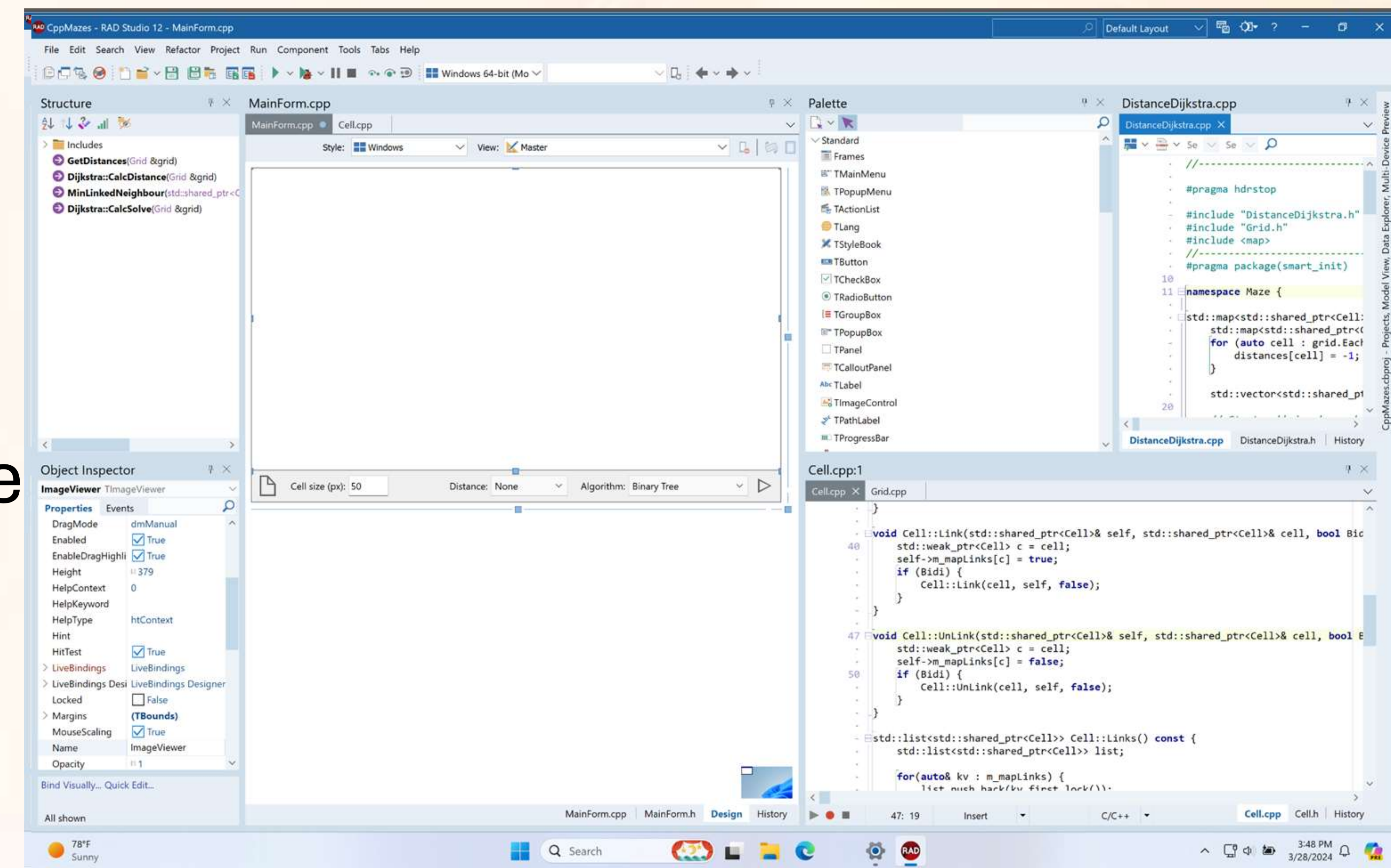
Editor split views

- Multiple editors (**tab groups!**) side by side
 - Or above/below each other
 - Or any combination!
- Split editors allow **same file** to be edited next to itself
 - Allowing you to edit the same file in multiple locations in code
 - Keep both **design & code** on screen
- Split editors allow many files in many **tab groups**, docked, windowed...



Why editor split views?

- Productivity:
 - Incredibly useful for coding
 - More than one editor in any IDE window
 - See two or more things at once
- Very useful for code and form design at the same time, great for the developers who miss the old undocked designer
- Usability
 - Focus is clear - solid blue bar



Additional IDE Quality

- \$(SanitizedProjectName) can be used in the project output path and the project final output directory
- \$(ProjectName) is now defined when building
- Saving desktops (and window positions) has been improved.
- The Data and Address Breakpoint dialogs have updated their UI to match the normal (source) breakpoint dialog.
- Non-visual components no longer show captions in data modules; the 'Show component captions' setting now works in data modules.
- An issue where IDE toolbars expanded to half the screen size has been resolved

Splash with multiple items

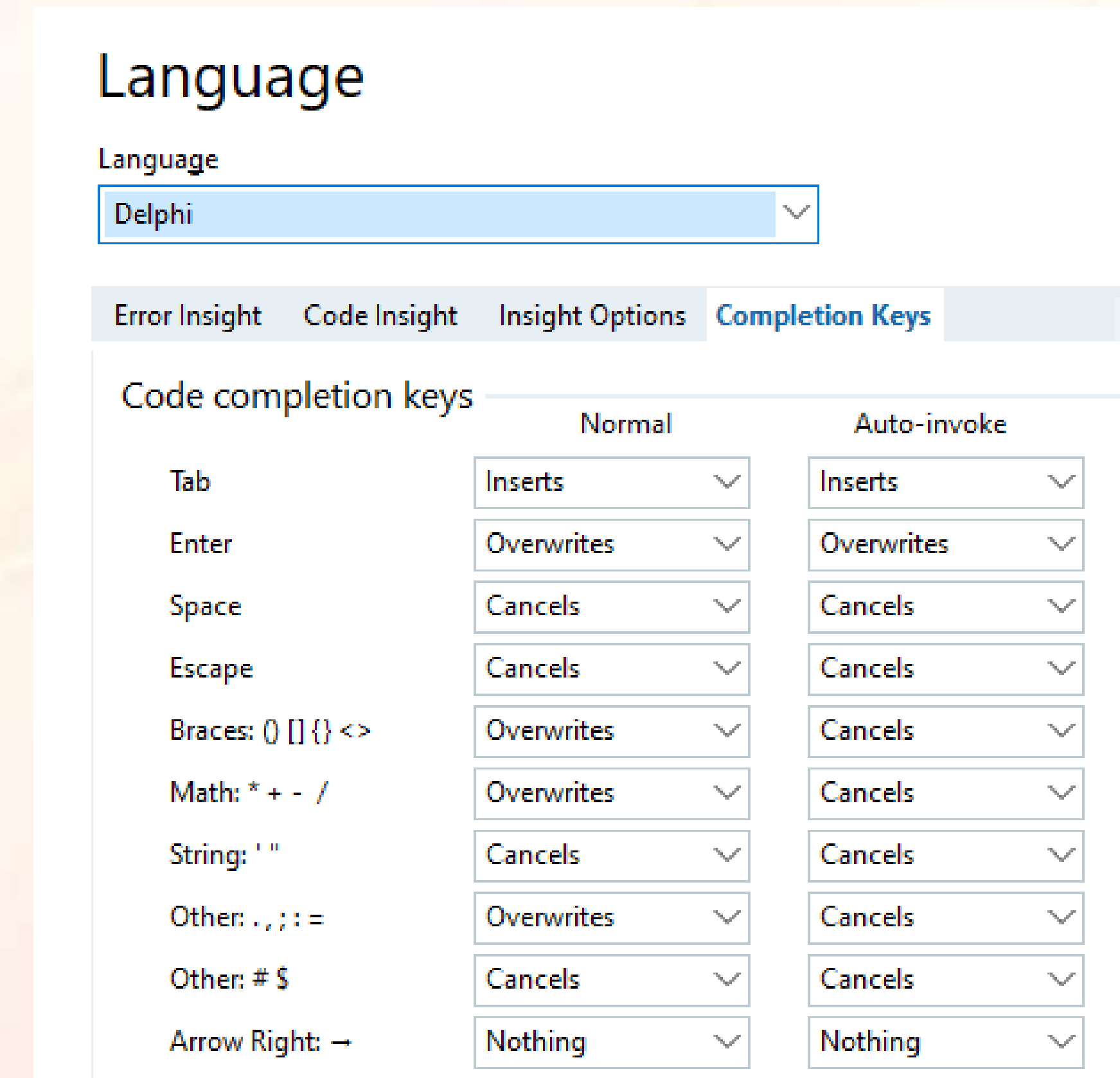


Today, this expands to three columns, and eventually a horizontal scrollbar.

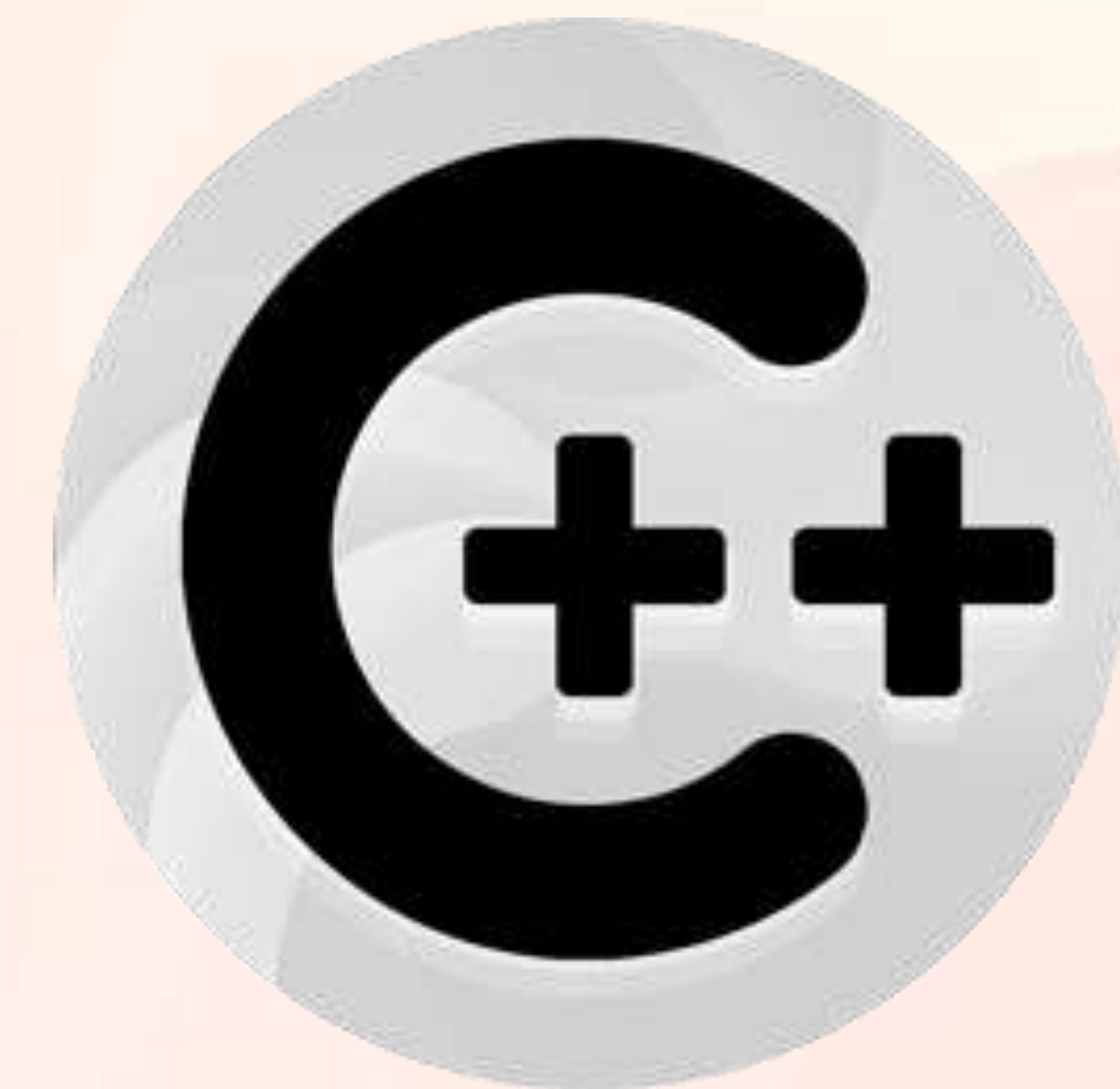
Delphi LSP Quality and Improvements

Goal: making auto-completion work smoothly and avoid anything unexpected

- Ability to customize the behavior of specific keystrokes or ‘*acceptance keys*’ when code completion is shown
 - When shown normally
 - When auto-invoked
- Customers can configure completion to match preferred behavior
- Added a number of improvements to Delphi LSP overall
 - Several fixes for class completion
 - Better code completion inside generic classes in more situations



Demo: Delphi LSP



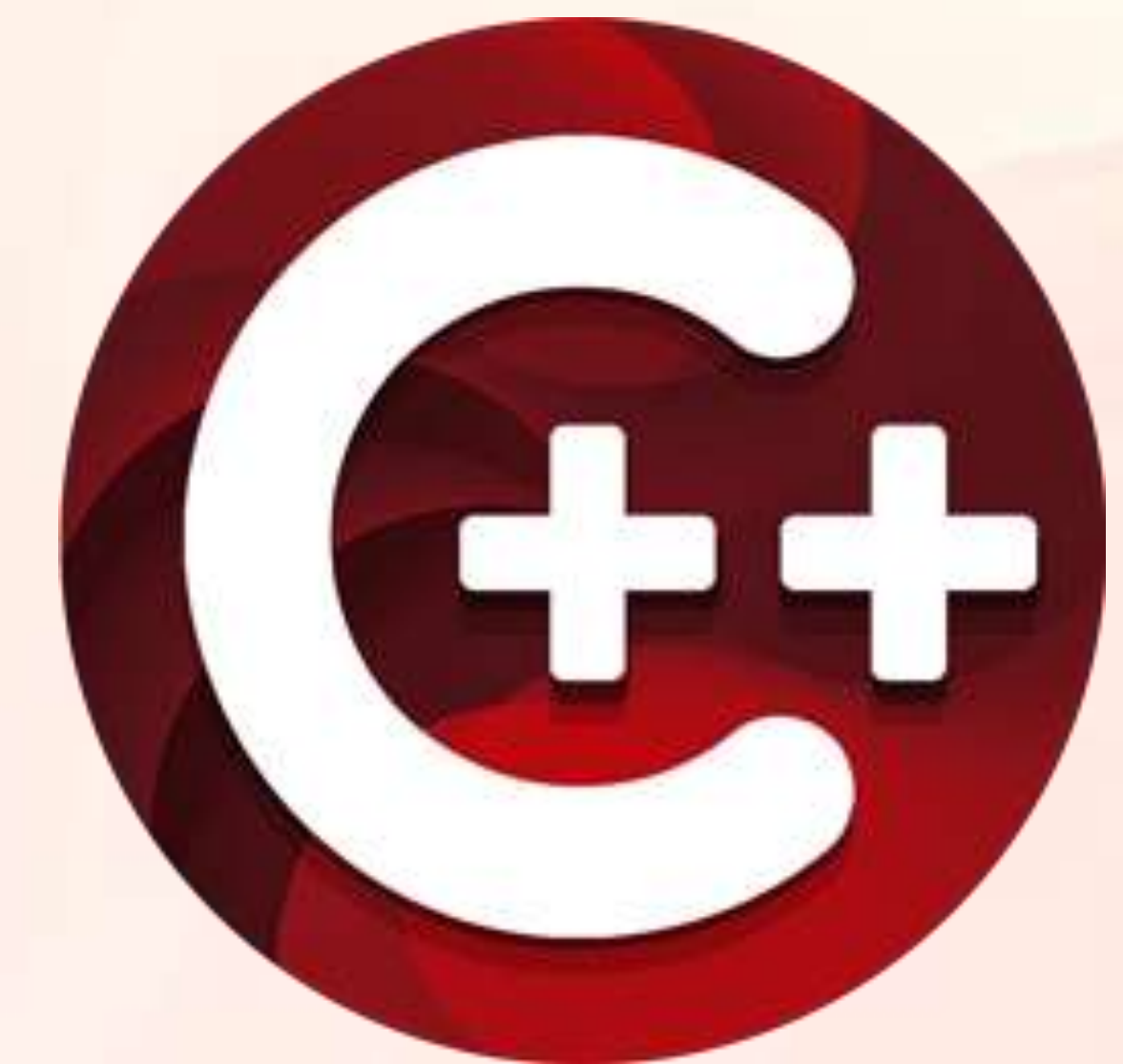
Visual Assist work in 12.1

Focused on quality

- Expanding rename refactoring to rename across code and visual designers
 - Renaming a component will include renaming it in the designer
 - Renaming now also renames event handlers
- Solved an issue where VA was left with a cache if a file was closed unsaved
- Code completion on object properties is working correctly
- Improved `#include` preprocessor directive autocomplete with `<` and `“”`
- Minor UI improvements on menus

A great feature for C++Builder 12. More to come in 12.x releases!

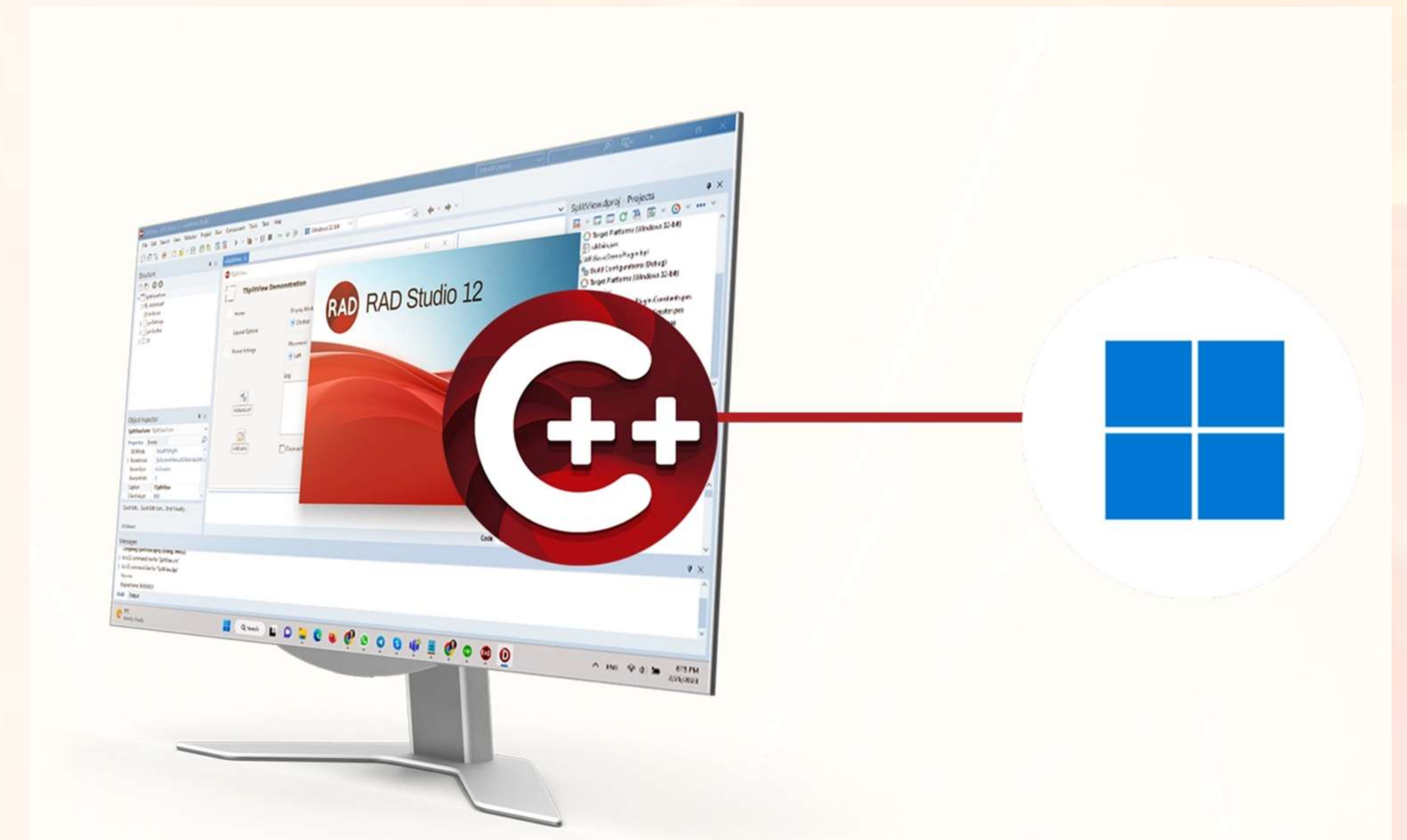
Demo: VA Improvements



Clang Upgrade in C++Builder 12.1

A new foundation for C++Builder and RAD Studio

- New version of Clang
- Revision of the entire toolchain
- Focus on *platform conventions*
- Focus on *doing it right*
- Focus on *quality*
 - STL, linker etc: a must-use



The value of C++Builder with a high quality toolchain for modern C++

Clang Upgrade in C++Builder 12.1

Feb 2024: Behind the Build for 12.1:
[youtube.com/watch?v=Ps5pW5uhmMw](https://www.youtube.com/watch?v=Ps5pW5uhmMw)

Nov 2023: released 12.0 with Clang Preview

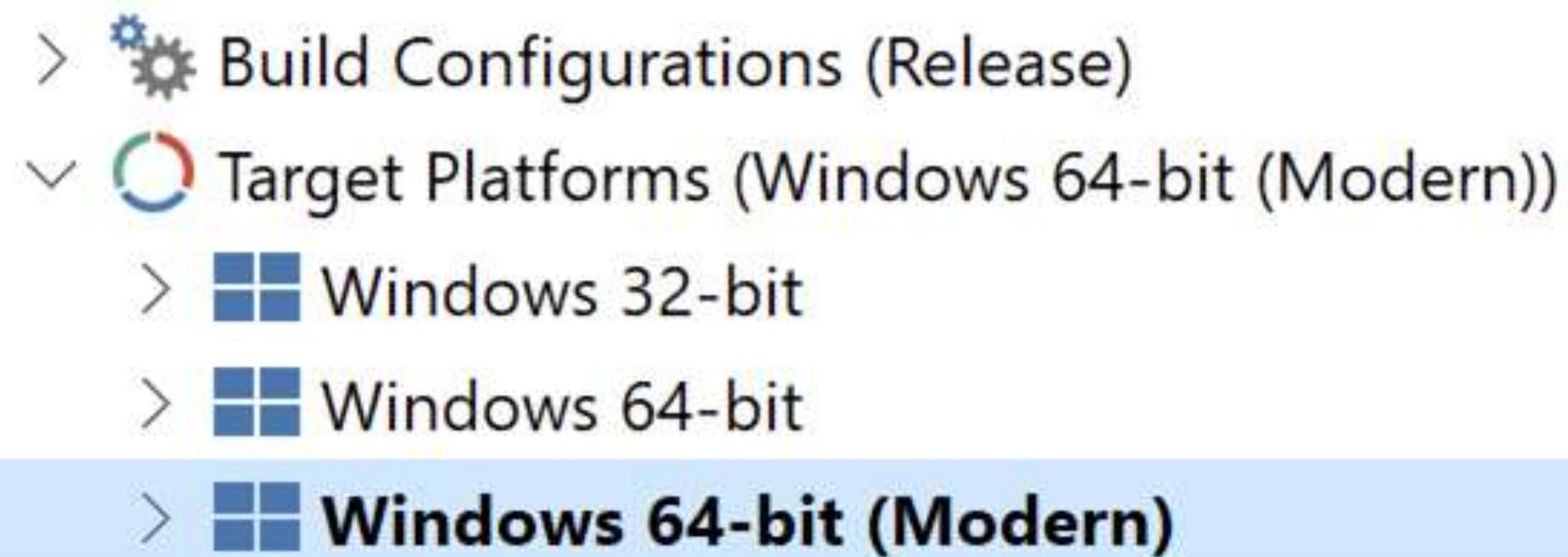
Sep 2023: Behind the Build for 12.0
[youtube.com/watch?v=B0Be_NFmEEE](https://www.youtube.com/watch?v=B0Be_NFmEEE)

March 2023: Blog, 'What's coming for C++Builder'
<https://blogs.embarcadero.com/whats-coming-for-cbuilder-an-amazing-preview/>



Clang Upgrade in C++Builder 12.1

- Shipping the first official version
 - Targets Win64
 - Existing Win64 compiler remains available in-IDE to help migration
 - Build your VCL and FMX apps
 - New toolchain: compiler, linker, STL, C++ RTL
 - Excellent compatibility with C++ standards and 3rd party libraries
 - Very exciting: performance, standards, quality



The value of C++Builder with a high quality toolchain for modern C++

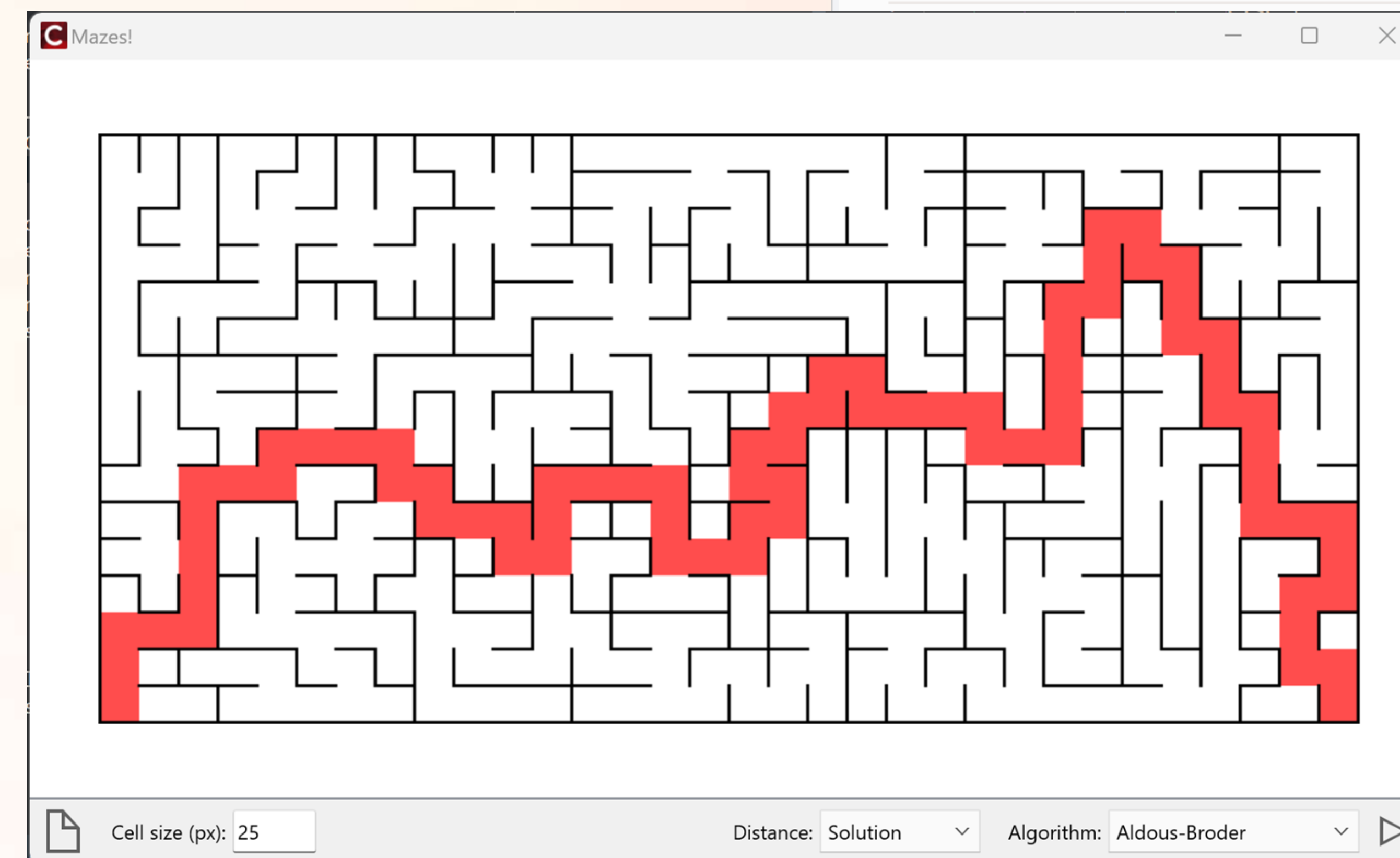
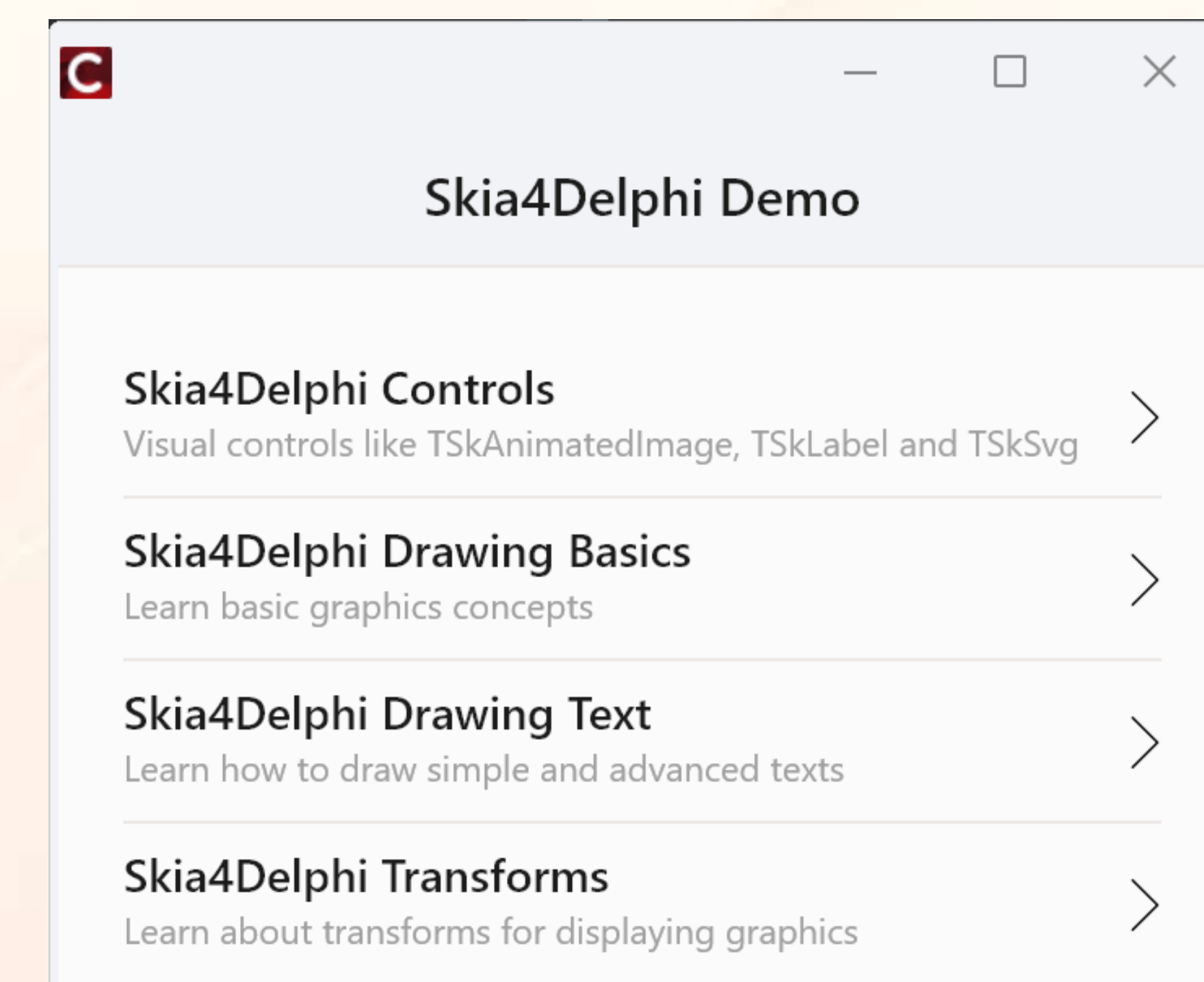
Technical Info of the new Clang Toolchain

- General goal: follow Windows platform standards
 - Leads to COFF, UCRT, PDB, etc
- Uses the Itanium C++ ABI
- COFF object file format, PDB debug format
 - Not link-compatible with a VC++ COFF object file, different C++ ABI
(goal: *source compatible*, handle any C++ code)
- C RTL: Uses the Windows component: Universal C Runtime (UCRT)
- C++ RTL, with enormous amount of exception handling work - 700+ new tests + more
- STL: LLVM's libc++
- Linker: LLVM's lld (used for Chrome)
- 64-bit binaries (even in IDE) – enormous memory space

Clang Toolchain: VCL and FMX apps

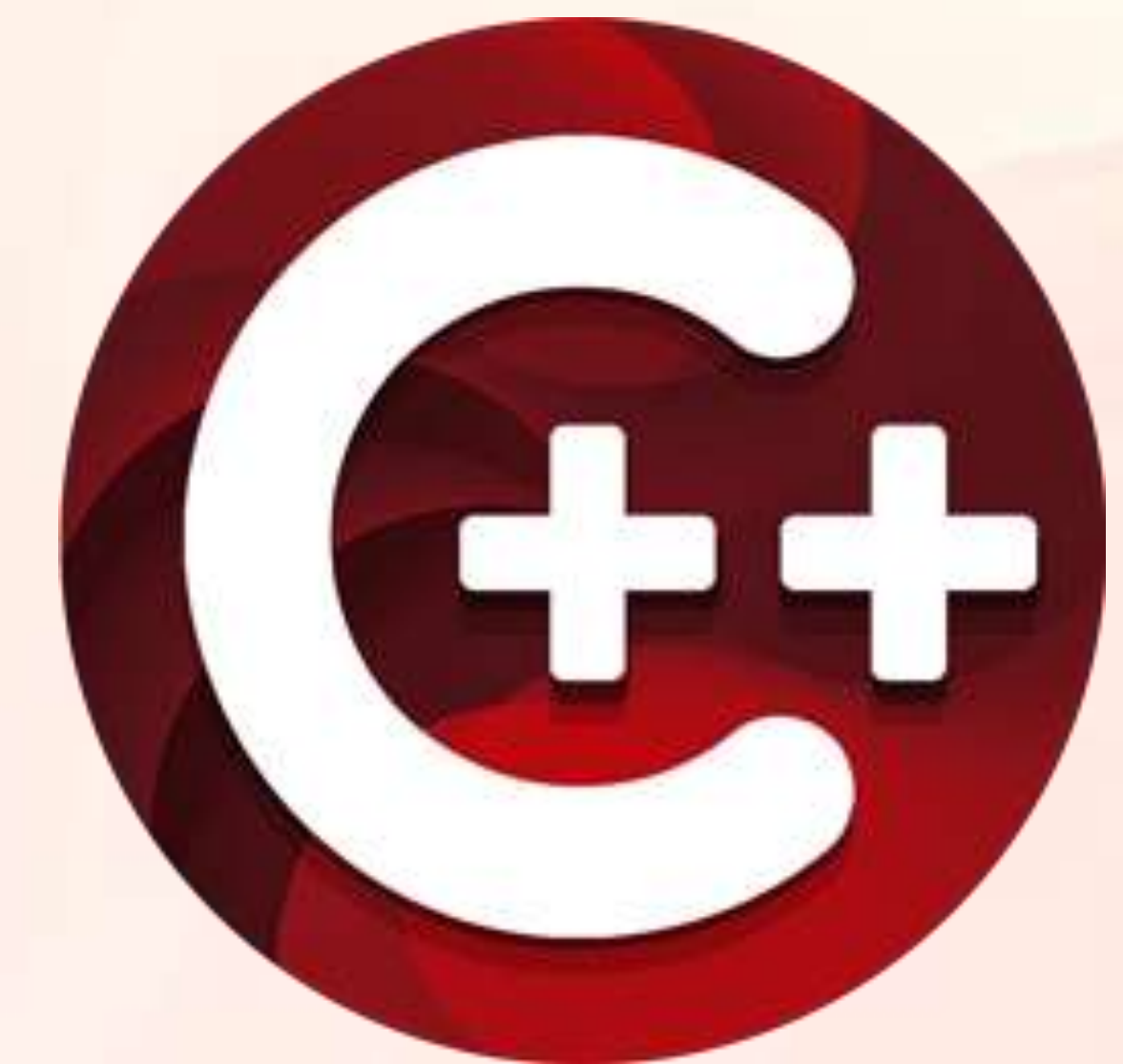
- VCL applications
- FMX applications
- DLLs, LIBs, console apps, etc
 - Can be 'pure C++' for these if you wish
- In 12.0, Delphi packages (components) are statically linked – but much faster linker to make up for it
 - I.e. the Delphi default! Never had a fast enough linker to do it before
 - These smaller features, like dynamic package linking and CMake support, coming soon

Skia in VCL used from C++



FMX 'Mazes' app

Demo: Clang Upgrade



Clang Upgrade: Why It Matters to You!

Quality

- Linker problems? (Eg memory issues) -> *Use this toolchain*
- STL problems? (Using 3rd party C++ code or modern C++ code?) -> *Use this toolchain*
 - *Much better STL! Things that didn't work before, do work now -> Use this toolchain*
- Compiler stability (ICEs) etc -> *Use this toolchain*

Compatibility

- Want to use other tools? (Eg other debuggers) -> *Use this toolchain*
- Use C++ code from anywhere -> *Use this toolchain*

Modernisation

- Clang 15, much more recent.

Performance

- Linker about 4x faster -> *Use this toolchain*

Feb 2024: Behind the Build for 12.1:

[youtube.com/watch?](https://youtube.com/watch?v=Ps5pW5uhmMw)

[v=Ps5pW5uhmMw](https://youtube.com/watch?v=Ps5pW5uhmMw)

Our foundation for C++'s future – here for you now.

Android API Level 34

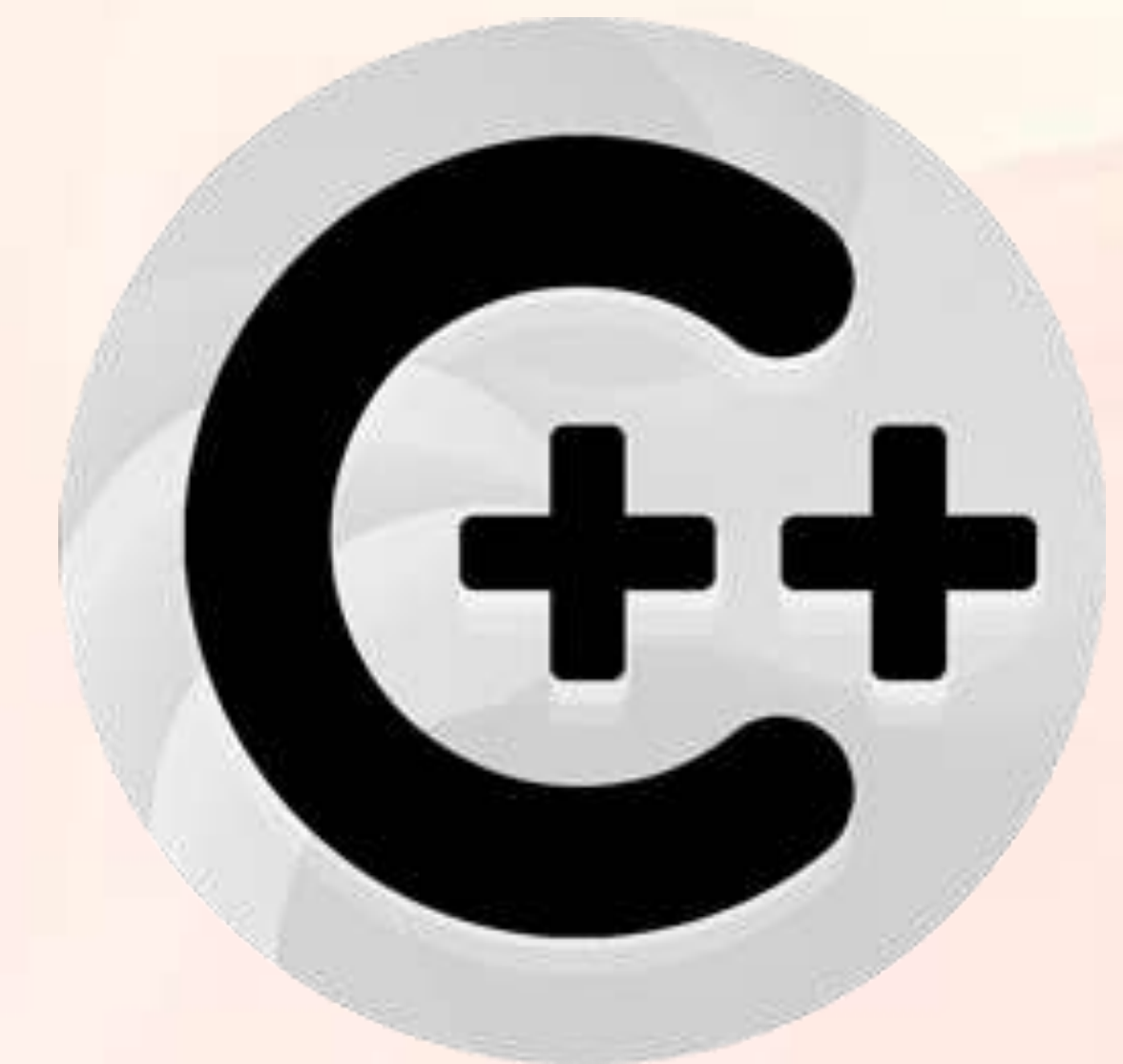
Android API level 34 support

- Updating the supported Android API to level 34
 - This has been done in advance of the August 2024 deadline for Google Play Store app submission
 - The default targetSdkVersion manifest attribute is now 34
- This change required (more technical details in next slide)
 - Updates to the Android SDK
 - Introduction of some additional Android platform tools
 - Update of the Java runtime
 - Improvements to the app packaging process
 - Updates to the uses permissions
 - For example: detect screen capture, foreground services

Android API level 34 support (more info)

- Eclipse Temurin JDK OpenJDK 17 (Hotspot) JVM
- Command-line Tools Android SDK components from Android SDK
- Improvements in the Android packaging:
 - Android DEX operations (DEX compilation and DEX merging) are now incremental via a new command-line tool called dexter
 - Use Google's AAPT2 tool (more control over the packaging process)
 - Packager command-line tool to package resource files, etc
 - New packaging project options
- Noticeable performance boost in the generation of .apk files
 - Due to the use of Google's zipflinger and signflinger libraries

Demo: Android API Level 34



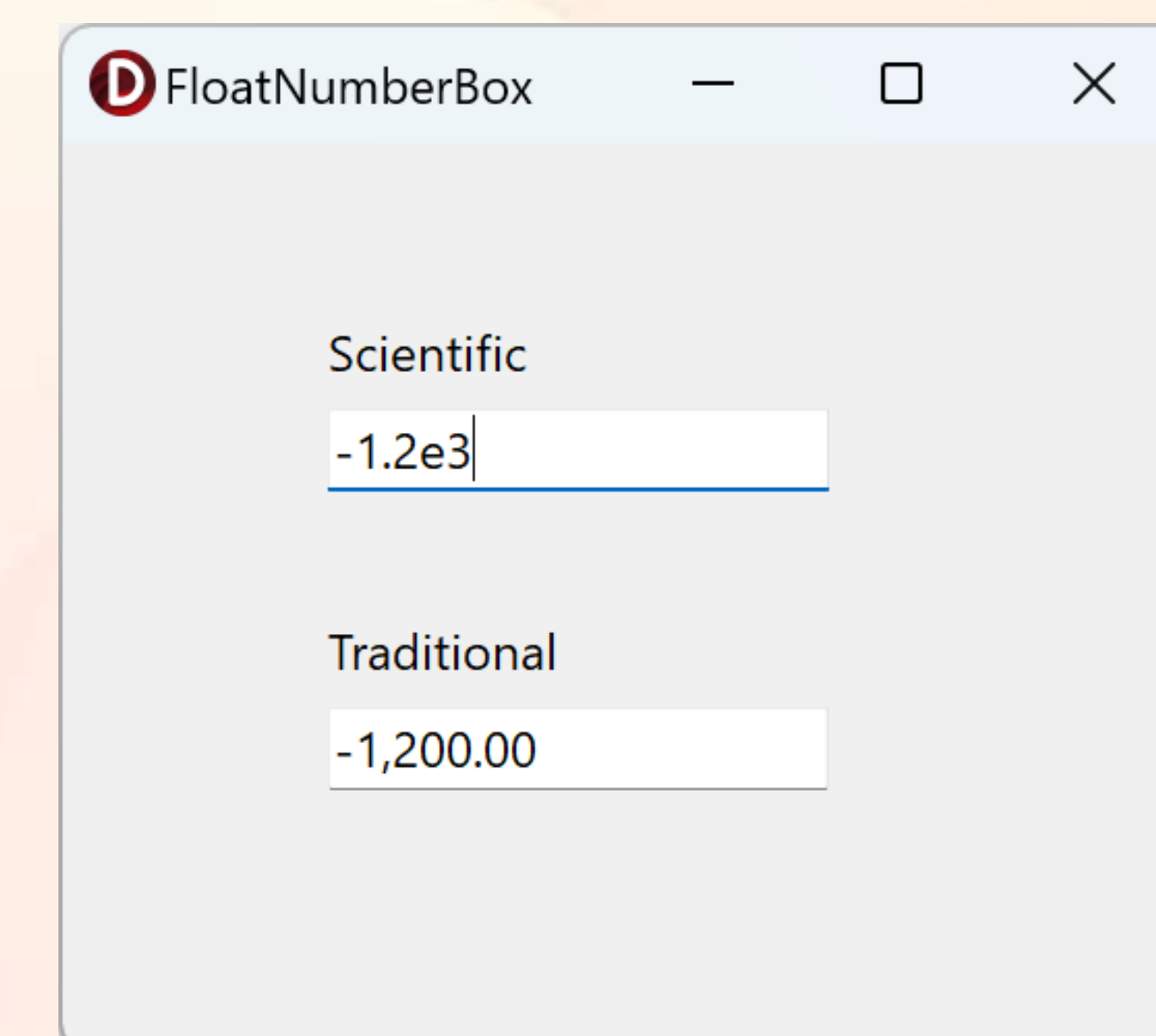
Android API level 34: Why Does it Matter to You?

- We are supporting a future Google requirement for the store
- We have made packaging apps and deploying them to Android higher quality and faster, using the most recent tooling from Google and the platform SDK – the same tools the “native” Android Studio uses

Libraries Quality

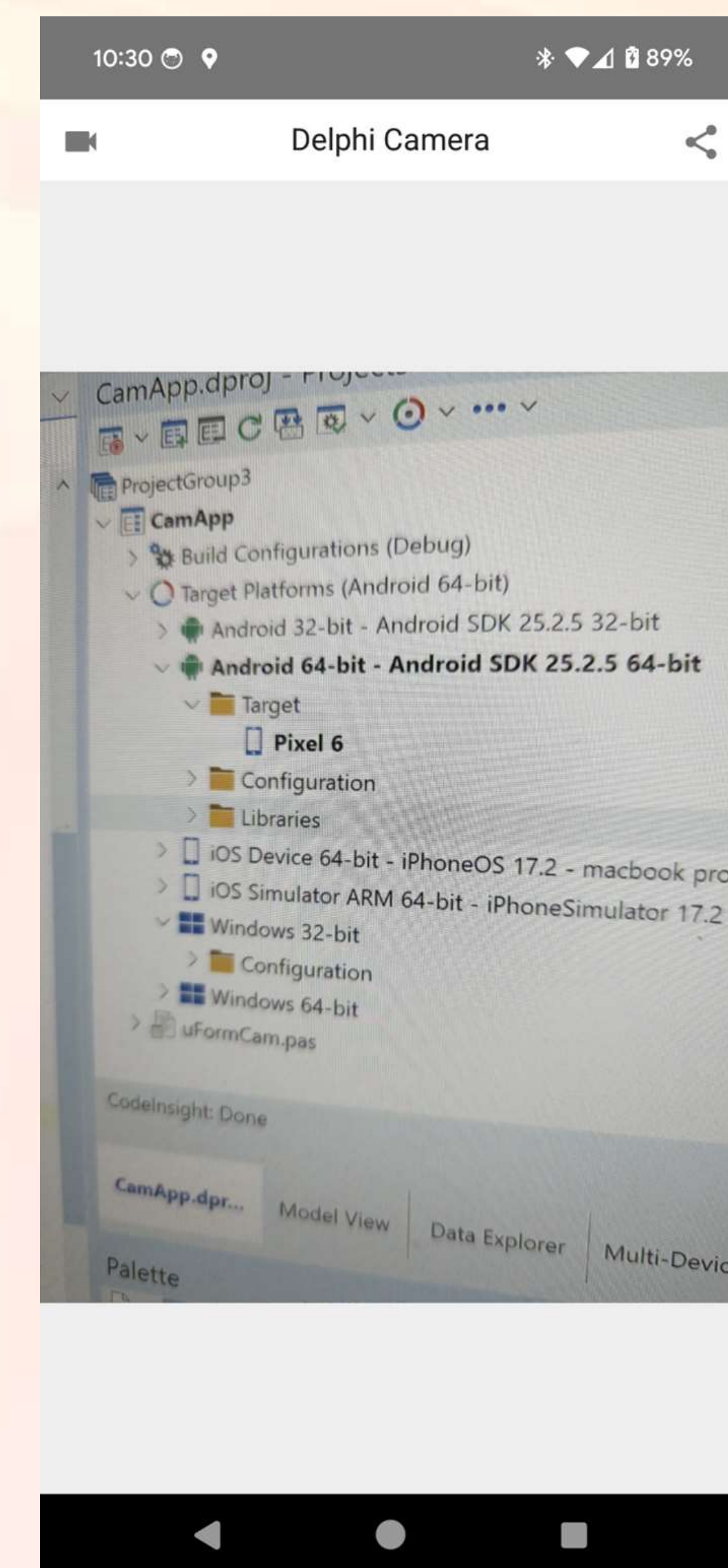
VCL quality

- Improvements for recent MDI redesign
 - Including MDI menu merge and styles, tiling, and more
- Enabled VCL TEdit to work with Windows *Ease of Access Text Cursor*
 - Re-enabled accessibility to the IDE
 - Using JAWS with the IDE now works by default
- NumberBox improvements
 - Better support for negative values
 - In nbmFloat mode, support for exponent, as in ‘-1.2e3’
- ProgressBar State is now supported under VCL Styles
- Addressed a few issues in Konopka Signature VCL Controls (KSVC)
- Additional work on styles, High DPI, TWICImage and much. much more



Extensive FireMonkey quality work

- On macOS, transparent bitmaps saved to non-transparent formats behave like on Windows (i.e., the transparent color will turn black).
- Further improvements for TMemo component (9 different extensions), TEdit (7 different extensions), FMX grids, FMX effects (with Skia), Keyboard shortcuts on desktop, the Flow Layout, Treeview, Listbox...
- Addressed issues in Android actions for library access and photo sharing, plus camera service



DataBase & FireDAC

- FireDAC adds official support for
 - Firebird version 5
 - Includes support for parallel operation in TFDFBBackup for Firebird 5
 - PostgreSQL version 16
 - InterBase 2020 Update 5 versions of IBToGo and IBLite for Android
 - SavePoint / RestorePoint support for a cloned TFDMemTable
- Database RTL
 - Cleanup in DataSet's OnValidate event
 - Some TDBNavigator fixes

REST and HTTP

- TRestClient supports using certificates from the local machine list
- Certificates support in TCurlHttpRequest
 - We recently introduced support for Curl also on Windows (beside Linux)
- Support for web hex color syntax in StringToColor function
- The ability for the REST Debugger to copy values from table cells
 - The REST debugger is a very nice, standalone, free tool by Embarcadero
- Ability to change the username of RAD Server users
- Improvements in JSON objects serialization and TJSONNumber decimal value handling with scientific notation

Overall Quality Stats

- Over 700 quality and performance enhancements, including:
 - 300 customer reported issues in Quality Portal addressed
 - Over 50 support tickets resolved
- Top focus areas (by number of tickets addressed)
 - IDE
 - FireMonkey
 - Compiler
 - VCL
 - RTL
 - Data



Live

Q & A

RAD 12.1 Summary: Focus on Quality

As we have seen in this session

- **Some substantial features**

- The delivery of the new C++Builder Win64 Clang compiler
- Editor split views and other IDE enhancements
- Android API level 34 support

- **Quality highlights include**

- C++ Builder VA integration
- VCL and extensive FireMonkey quality
- RTL, database, etc
- New Clang also heavily focused on quality!
- Includes the 12.0 Patch 1 Fixes
(released in January)



New Embarcadero Quality Portal

- Announcing the new portal:
<https://embt.atlassian.net/servicedesk/customer/portals>
- Powered by Jira Service Manager
- Integrated with the Embarcadero R&D Jira instance
- Hosted by Atlassian Cloud
- Self service accounts
- New Feature requests, Bug reports and Documentation Bug reports

