



Introducing

RAD Studio 11.2 '*Alexandria*'

embarcadero®

Today's Presenters



Fernando Rizzato

Lead Software Consultant
Embarcadero

fernando.rizzato@embarcadero.com

@FernandoRizzato

The Landscape

- **Microsoft released Windows 11**
 - Fully focused on it, key reason for picking "11"
 - New dev platform Windows App SDK
(aka *Project Reunion*)
- **Transition to ARM on desktop**
 - Led by Apple with macOS
 - Plus, ARM reigns on mobile platforms
- **Developers on 4K**
 - High resolution and multi screen are common for developers
 - Growing also for end-users





What's Great in RAD Studio

Developer Productivity

Shipping is a feature. Get to market 5x faster with Delphi and C++Builder

Fast Native Apps

With native compilers, RAD Studio gives your apps the speed they need (no bloat)

Database Access

Key to Delphi's initial design, database access is integral to RAD Studio

Powerful C++ Libraries

Hundreds of C++ libraries can be used in C++Builder or Delphi in RAD Studio

Platform API Access

Gives you access to all the platform APIs on all platforms

Visual Designers

Forget prototyping in another tool, just design it and finish it in the RAD Studio IDE

Strong Community

Full of Technology Partners, MVPs, trainers, authors and developers

Backward Compatibility

You have an investment in your code, RAD Studio keeps that relevant



RAD Studio 11.0 Summary

Large Focus on IDE

- IDE High DPI and High DPI enabled form designer
- VCL styles in VCL designer; improvements to FMX designer and Multi-window design/code for a unit
- LSP quality and new C++ code formatter
- New Welcome page
- Visual Studio Code LSP support
- Minimal installer improvements

Delphi and C++ compilers and toolchains

- C++ Quality: RTTI (components); exception handling
- Delphi macOS 64 ARM (M1) compiler and toolchain
- Minor Delphi language improvements

Libraries Improvements

- Delphi RTL quality: TZipFile, 64-bit data structures, Bluetooth LE
- VCL Modernization work
- Better FMX High DPI on Windows and desktop integration, including WebView 2
- Android API 30 support and related work

Additional Delphi Platforms

- Support for WSL 2 (Windows Subsystem for Linux)
- Including WSLg (Linux GUI) on Windows 11

RAD Studio 11.1 Summary

Focus on Quality and Improvements

- Usability, performance, stability
- Some new small IDE features

Better Code Insight and LSP

- For both languages, Delphi and C++

Platforms and Tooling

- Official support for Windows 11, macOS 12 Monterey, iOS 15, and Android 12

Libraries

- Performance and quality improvements and small additions to Delphi RTL, database libraries, VCL, FireMonkey

RAD Studio 11.1.5 (C++) Summary

C++-only release for Code Insight and LSP

- Performance, solving quality issues, and more
- Also related changes around saving projects



RAD Studio 11.2: iOS Simulator & Quality / Polish

Focus on Quality and Improvements

- IDE and Designer High DPI
- Usability / 'The Small Things' Quality

IDE Improvements

- CPU View Syntax Highlighting
- Markdown Support
- Editor Tab Improvements
- Fonts and Styles

Code Insight / DelphiLSP

- Reworked Help Insight (and using XSLT again)
- Marking Inactive Code and Color Constants
- C++ Code Insight from 11.1.5

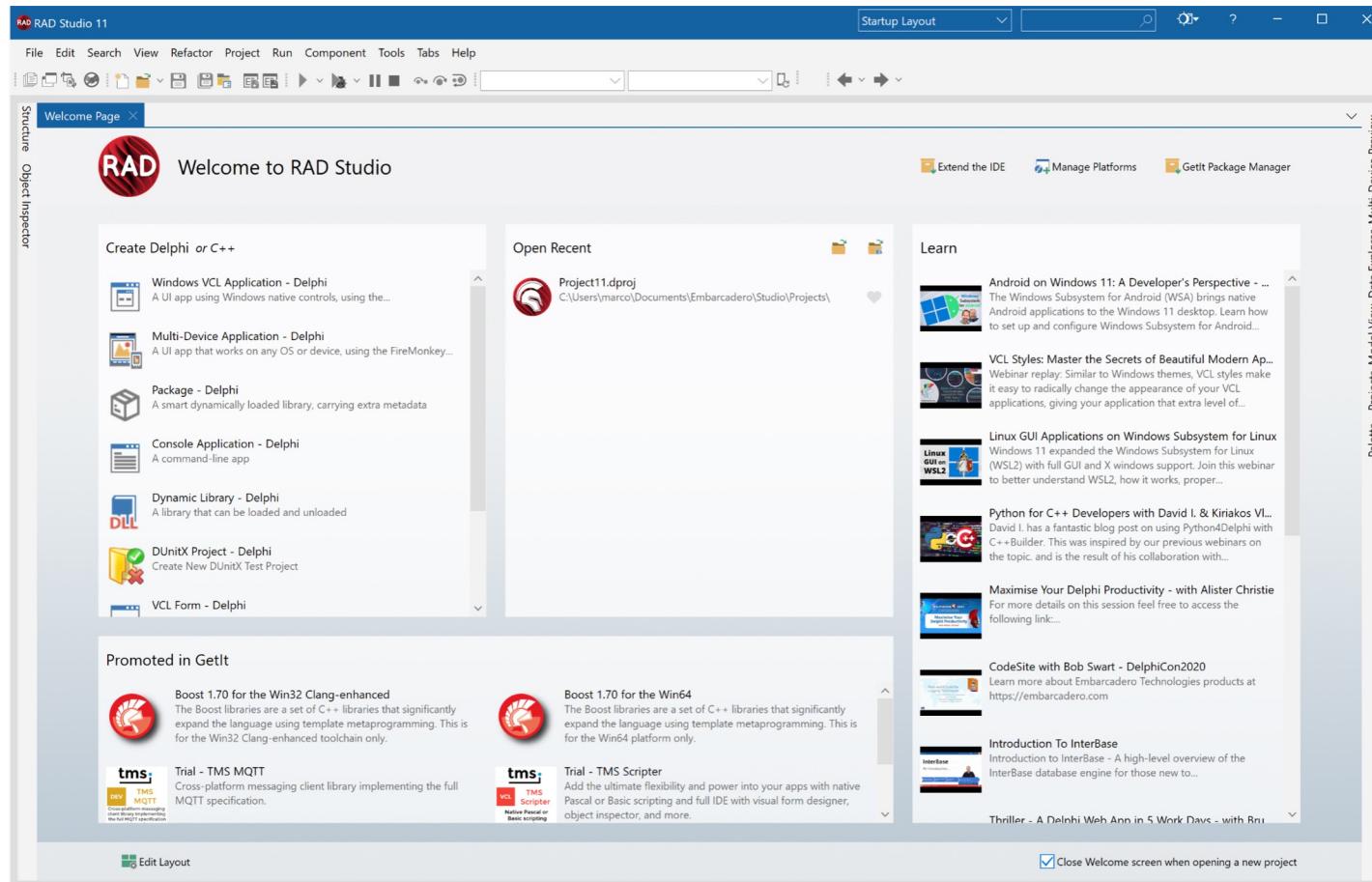
Platforms and Tooling

- iOS Simulator for Delphi
- LLDB for Linux
- Android Target API 32
- Remote Debugging Win64 C++

Libraries

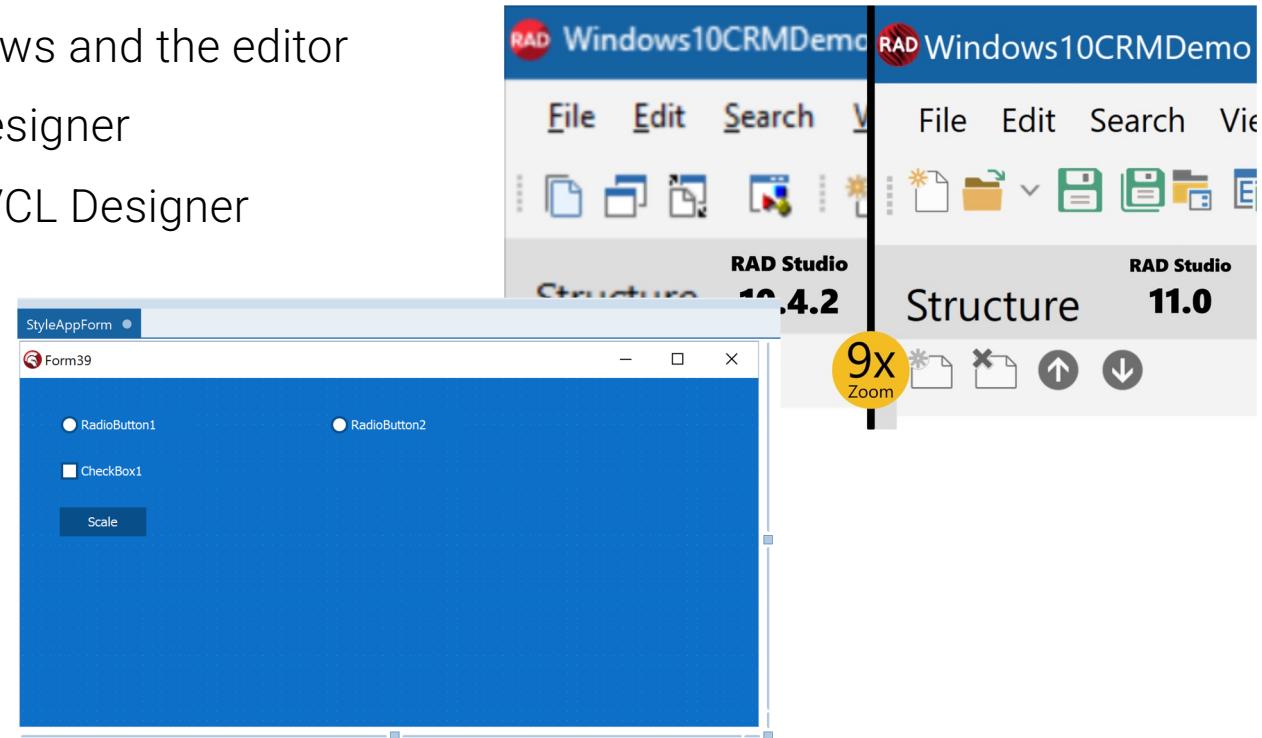
- VCL WebView2 Improvements
- General VCL and FMX Quality
- ZLib, HTTP / REST
- RAD Server
- FireDAC reFind, Updated Drivers, and FDMonitor/FDExplorer

I. RAD Studio IDE



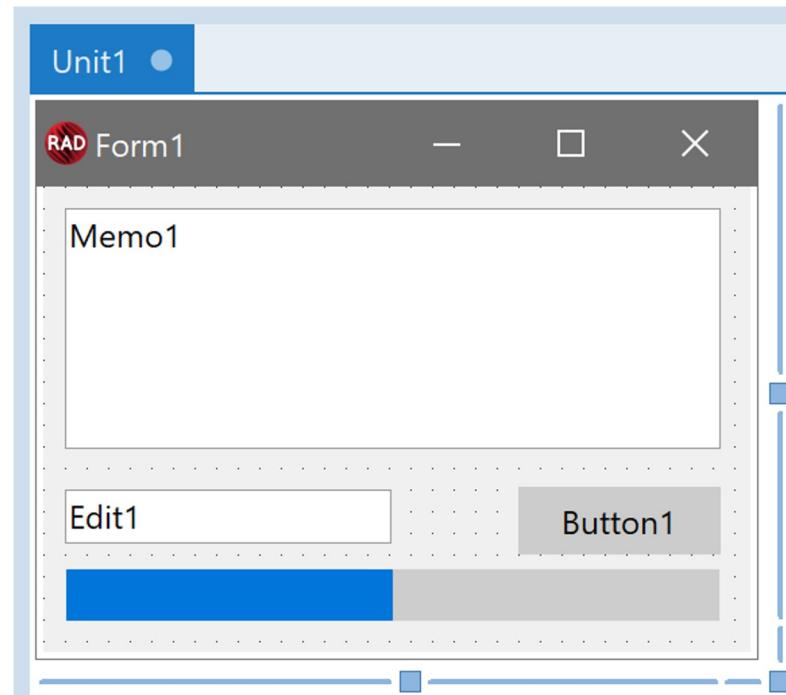
11.0 and 11.1: IDE and Designers

- High DPI IDE
 - Crisp fonts in all windows and the editor
- High DPI enabled form designer
- Multiple Active Styles in VCL Designer



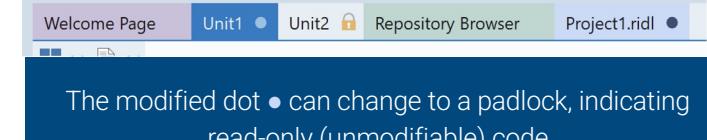
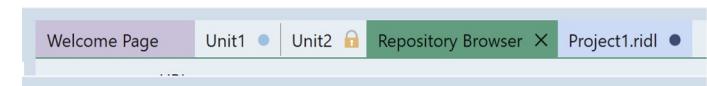
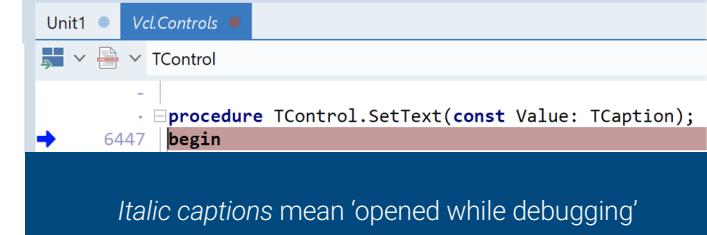
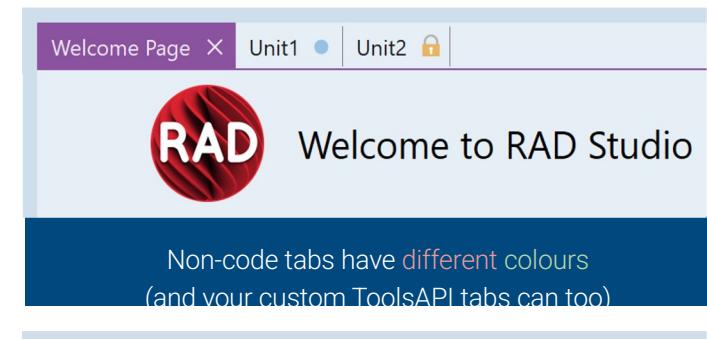
11.2: IDE Designers

- High DPI Quality improvements
 - Many areas, many windows: overall quality
 - VCL Designer has many High DPI fixes
- VCL Designer uses styles by default
 - In 11.0, added optional (off by default) styles in the designer, including multiple styles at once, just like apps
 - In 11.2, this is on by default, and a special design style is used at designtime



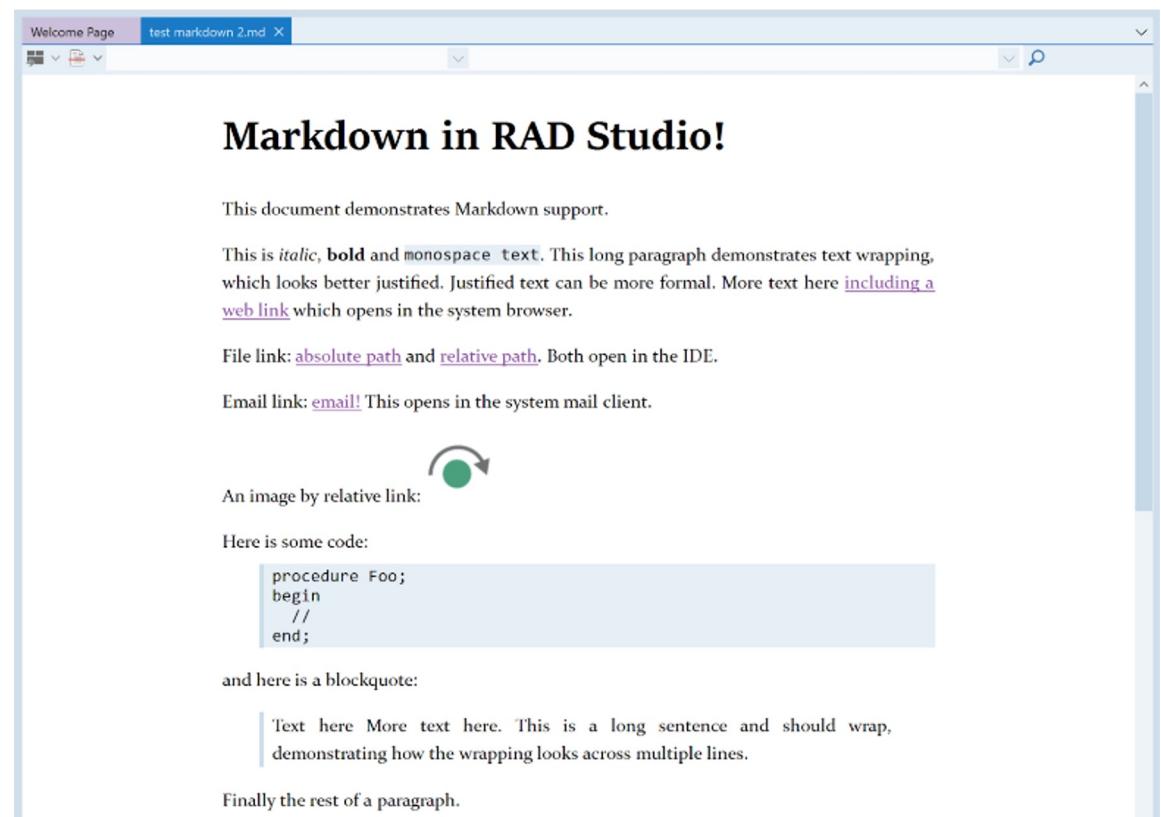
11.2: IDE - tab state visualization

- Some tabs are different to other tabs:
 - Some have *different, non-code functions*, eg the Welcome Page, Type Library editor, source control tabs
 - Some code is *opened while debugging*
 - Some code is *read-only*
- In the past, we indicated modified code with a dot in the tab
 - **In 11.2, we add symbology for all the above as well.**
Helps visual navigation
- Plus, the X to close is now configurable: off entirely, on the active tab, or on all tabs



11.2: Markdown and HTML

- Markdown is a common human-readable ‘markup’ (formatting) format
- Most GitHub readmes are Markdown
- IDE supports Markdown documents, richly formatted: headings, links, images, code, etc
- Rich HTML returns as well
- HTML and MD editable in code editor
- Project readmes can be Markdown
- No Internet Explorer in the IDE any more!
Pure VCL implementation



11.2: Help Insight

- The return of XSLT!
 - Effectively, Help Insight as it was pre-11.0
 - Full XmlDocument info
 - Can be customised by editing XSLT file (eg, as Documentation Insight does) and CSS
- Contains links to navigate

```
if (GetWindowLong(Handle, GWL_EXSTYLE) and WS_EX_TOPMOST <> 0) and
(
  function Winapi.Windows.GetWindowLong(hWnd: HWND; nIndex: Integer): Integer - Winapi.Windows.pas \(42293\)
(
  Parameters
    Ap
      hWnd
      HWND
    El
      nIndex
      Integer
  Re
end;
```

11.2: Delphi: see Inactive Code in the editor

- A common customer request!
- Not all code is compiled:
it is often {\$IFDEF ...}-ed out.
- Code that is not compiled = inactive
 - Now, inactive code will be drawn in a faded semi-transparent color
- Other markup, like matching braces or selections, still drawn for inactive code - but also a different color
- This can be turned off or the transparency edited in the registry

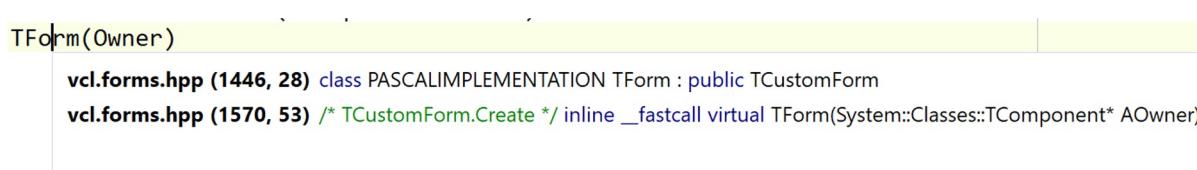
uses

```
FMX.Platform,  
{$IF DEFINED(ANDROID) OR (DEFINED(IOS))}  
FMX.DialogService.Async  
{$ELSE}  
FMX.DialogService.Sync  
{$ENDIF}  
;
```

```
if Addr = 0 then  
  Result := 'Nil'  
else  
{$IFDEF AUTOREFCOUNT}  
  Result := Format('%s@%8x(%d)', [ClassName, Addr, Instance.RefCount]);  
{$ELSE}  
  Result := Format('%s@%8x', [ClassName, Addr]);  
{$ENDIF}
```

11.2: C++ CodeInsight (incl from 11.1.5)

- From the 11.1.5 C++-only release, all included into 11.2:
- Improvements to save order of projects and files
- Performance (one test was 20 seconds -> 2 seconds to show the completion dropdown)
- Improved navigation (ctrl-click) including showing multiple destinations
- Third indexing option: index all files in all projects
- Many quality fixes incl for classic
 - We **strongly recommend** reading the Docs on configuring C++ Code Insight for your project setup:
 - Code Insight Reference > 'Index for C++ Insights' heading > 'LSP Behaviour'



TForm(Owner)

vcl.forms.hpp (1446, 28) class PASCALIMPLEMENTATION TForm : public TCustomForm

vcl.forms.hpp (1570, 53) /* TCustomForm.Create */ inline __fastcall virtual TForm(System::Classes::TComponent* AOwner)



11.2: CPU Disassembly View

- When stepping into assembly code, syntax highlighting can help make it much easier to read
- Match editor or IDE style colours

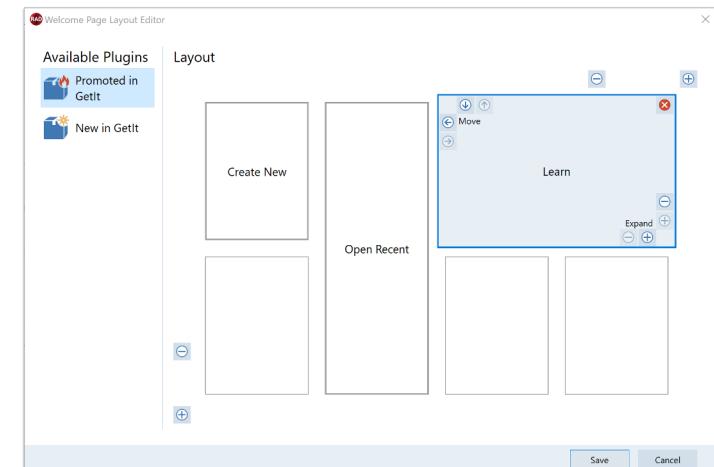
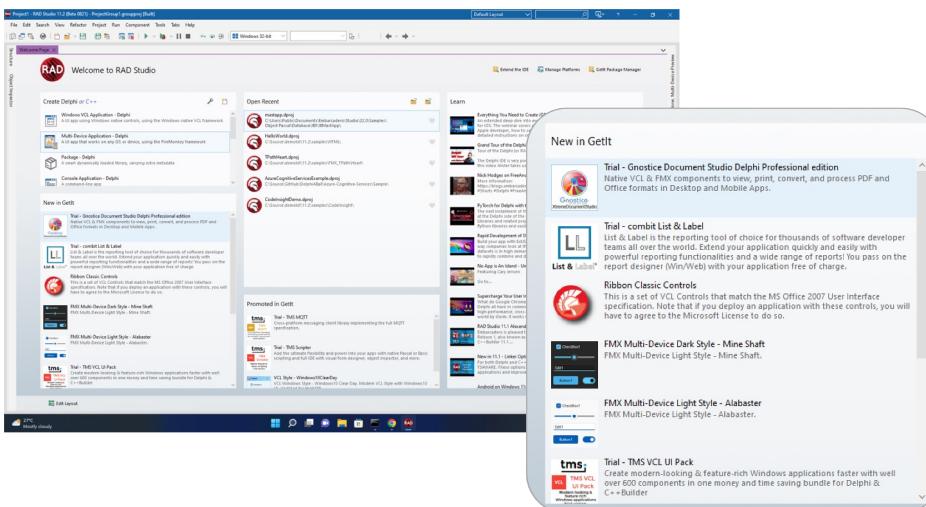
```

Unit1 CPU Vcl.Forms
Thread #7080
win32u.NtUserWaitMessage:
76D610C0 B80C100100    mov eax,$0001100c
76D610C5 BA1063D676    mov edx,$76d66310
76D610CA FFD2          call edx
+ 76D610CC C3           ret
76D610CD 8D4900         lea ecx,[ecx+$00]
win32u.NtUserTranslateMessage:
76D610D0 B80D100000    mov eax,$0000100d
76D610D5 BA1063D676    mov edx,$76d66310
76D610DA FFD2          call edx
76D610DC C20800         ret $0008
76D610DF 90             nop
win32u.NtUserGetProp:
76D610E0 B80E100000    mov eax,$0000100e
76D610E5 BA1063D676    mov edx,$76d66310
76D610EA FFD2          call edx
76D610EC C20800         ret $0008
76D610EF 90             nop
00410000 00 00 00 00 00 00 00 ..... 005FFD30 0001
00410008 00 00 00 00 00 00 00 ..... 005FFD2C 0001
00410010 00 00 00 00 00 00 00 ..... 005FFD28 0241
00410018 00 00 00 00 00 00 00 ..... 005FFD24 0001
00410020 00 00 00 00 00 00 00 ..... 005FFD20 0001
00410028 00 00 00 00 00 00 00 ..... 005FFD1C 0001
00410030 00 00 00 00 00 00 00 ..... 005FFD18 0011
00410038 00 00 00 00 00 00 00 ..... 005FFD14 00F20BDB .ó.U
00410040 00 00 00 00 00 00 00 ..... 005FFD10 005FFD38 ._ý8
00410048 00 00 00 00 00 00 00 ..... 005FFD0C 030FCE10 ..Í.
00410050 00 00 00 00 00 00 00 ..... 005FFD0E 00000000

```

11.2: Welcome Page

- In 11.1: New native Welcome page, VCL based and flexible.
Layout and content is fully customizable: show what and where you want
- New in 11.2: New in GetIt pane, shows recently added packages
See what's new – discover packages





11.2: GetIt improvements

- Language filter: search for only Delphi or C++
- Local packages (with update subscription)
 - Download a package, install on a computer not connected to the internet
 - Experiment with creating packages
 - Simplify team installation of libraries for multiple devs
- Will wait for an external installer

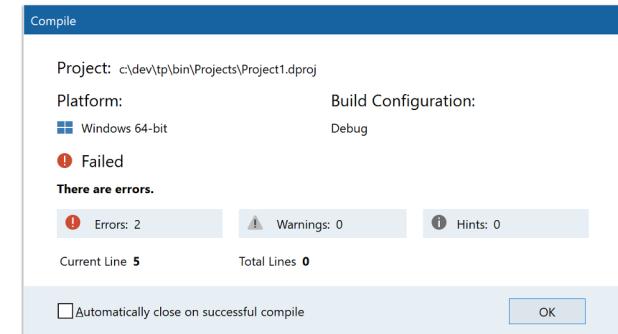
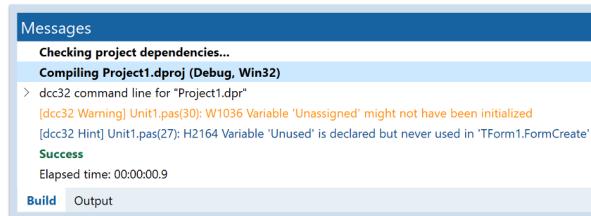
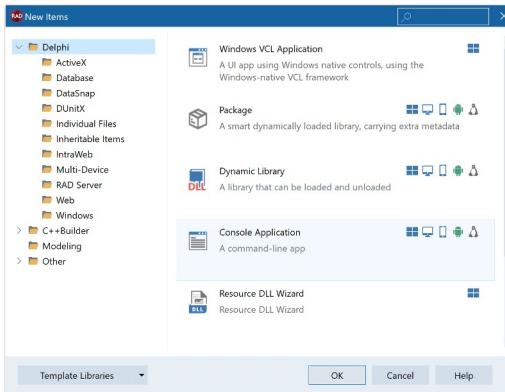
The screenshot shows the RAD GetIt Package Manager interface. On the left, there are filter options for 'Filter' (All, Installed, Subscription Only, Updates), 'Languages' (All, Delphi, C++), 'Sort By' (Name, Vendor, Date), and 'Categories'. The 'Categories' section lists various categories with their counts: All (346), Libraries (47), Components (71), Internet Of Things (51), Trial (67), Industry Templates (4), IDE Plugins (9), Styles (41), C++ Libraries (14), Tools (10), Sample Projects (29), and Promoted (3). At the bottom, there are buttons for 'Load Local Package' and 'Package'.

Category	Count
All	346
Libraries	47
Components	71
Internet Of Things	51
Trial	67
Industry Templates	4
IDE Plugins	9
Styles	41
C++ Libraries	14
Tools	10
Sample Projects	29
Promoted	3

Other IDE Enhancements in 11.x

- New Items dialog: available platforms for each item
 - Also uses TControlList and offers other views
- 11.2: File > New, now shows "Data Module" as standard

- The Compile dialog shows the platform and build configuration

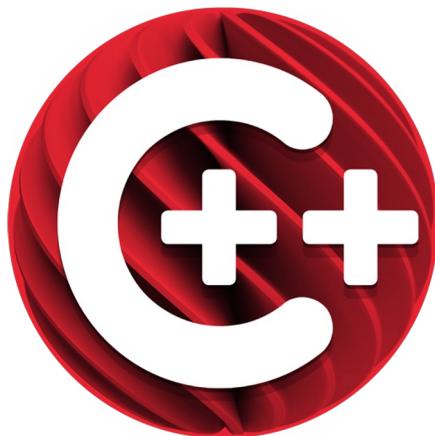


11.2: Polish: the little things that help

- RAD Studio 11.2 is full of small tweaks that improve quality of life
- *Some already mentioned are:*
 - Editor tabs, colours and opened while debugging
 - File > New > Data Module
 - VCL designer - high DPI improvements (control size, checkboxes and radios, etc)
 - Readonly units, padlock icon
- *Some not yet mentioned are:*

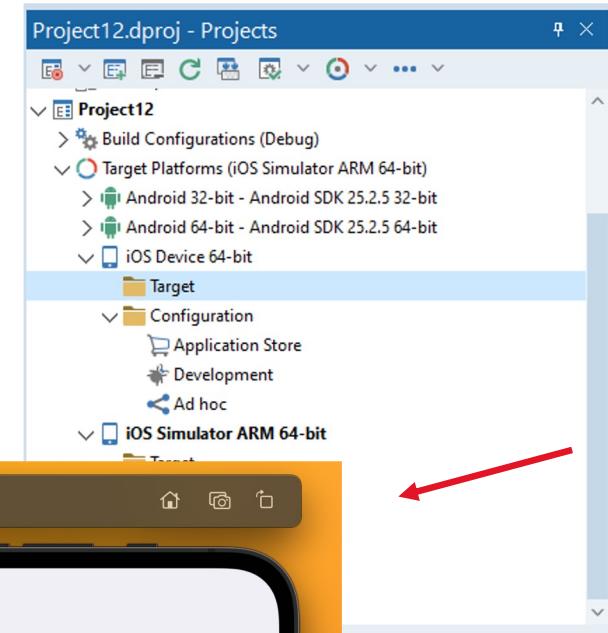
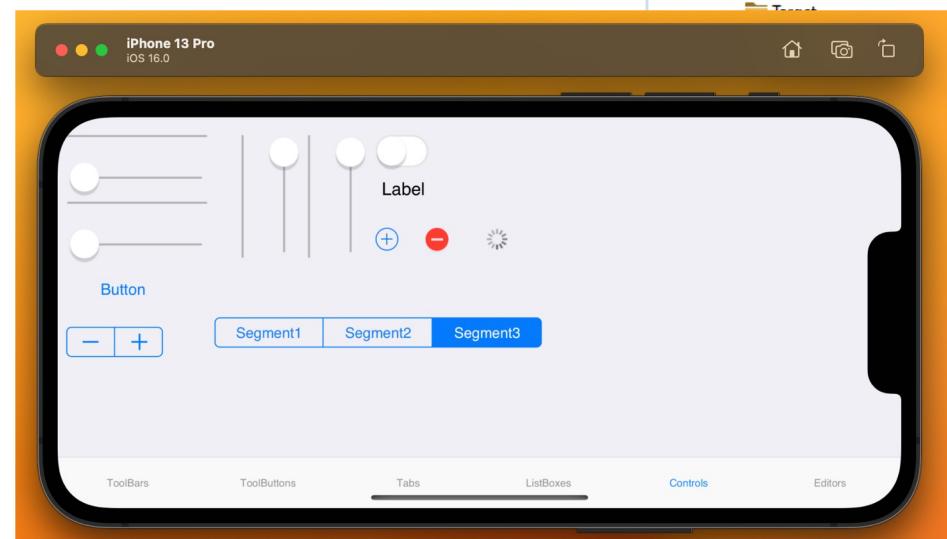
RAD Server, see IP address and more; [Projects double click: does what you expect](#); Reopen list always up to date; Compile and File Cache dialogs [format counts follow locale](#) and are easier to read; IDE performance when drawing / moving windows greatly improved; [Large number of UI layout, control rendering, control sizes, etc improved](#); IDE style shows the default button better; [Large font size support in the IDE: set in Options dialog \(not registry\) and supported for more windows](#); Custom style support in the IDE (any VCL style) can be set in the Options dialog; [Remote Desktop usage improvements](#); About box shows installed patches; [Delphi code completion shows color constants with their colour](#); **...and more!**

II. C++ and Delphi Compilers and Tools



iOS Simulator Support

- New Target Platform for Delphi in 11.2
- Simulator very useful for fast iteration:
 - Deploy to the simulator, and debug, is much faster than to the actual device
- Apple moving to ARM across all devices, including running iOS apps on Mac
- In 11.2, run Delphi iOS apps through the simulator on M-series CPU Apple devices



Delphi compiler: Quality Focus

- 11.x features plus all hot-fixes
 - Includes AtomicCmpExchange[128] fixes from 11.1 patch 1; 'set of' quality on LLVM compilers, and improvements to Win64 assembly instructions

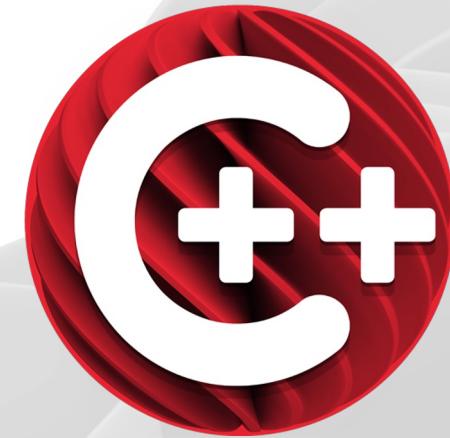
Security flags

- ASLR support extended with High Entropy Address Space Layout Randomization (HE-ASLR) in Win64
 - A good additional security feature. DEP/NX, ASLR (both in 11.1) and HE-ASLR often required for some industry areas
- Expect issues if you convert handles to ints: do not assume handles are 32-bit despite the common Windows norm, not all are.
- All our packages support it



C++ toolchain: Quality Focus

- Includes all work from C++-only release, 11.1.5
 - Code completion and navigation, as seen earlier
- STL iostreams performance: < 25% of 11.1 time, plus other quality improvements: integrated record helpers from Delphi, delay load improvements, linker fixes, data alignment IDE <-> bcc32c, and more
- Linker supports ASLR, DEP/NX, and TSAWARE flags (but not >4GB start address)



LLDB for C++

- Remote debugging for Win64!

Android

- Using Android API level 32
 - Play Store mandates level 31 starting Aug 2022 for new submissions, Nov for updates
 - Updates targetSdkVersion, AndroidManifest.xml template, etc
- JDK version is now v11
 - and if Eclipse Adoptium JDK is available, will be used in preference to others
- Improvements for local notifications
- Many new Uses Permissions
- Can also create a PKCS12 keystore



LLDB Update

Goal: add Delphi language support to LLDB so can evaluate expressions in Delphi in Evaluate/Modify, watches, etc, and move to LLDB across all platforms.

This:

- reduces multiple debuggers to one;
- gives consistent features across all platforms;
- and uses a modern debugger

Have seen this in past release webinars, and over (some) future releases, will continue to see progress on LLDB

11.2 extends Delphi support:

- basic handling of set expressions, dynamic arrays, 'nil', pointer casts, empty strings and strings with hex characters

LLDB Update

11.2: Linux 64 moved to LLDB

Requires Python 3, see docs if you have issues

As of 11.2, LLDB is used for:

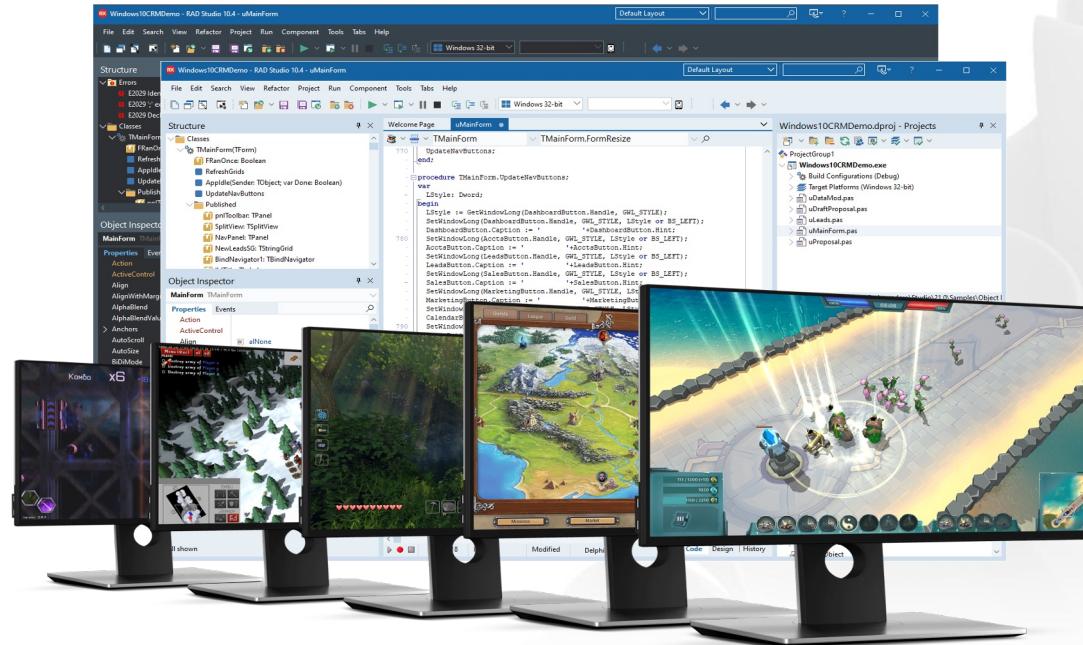
- C++ Win64
- Delphi macOS ARM 64
- Delphi macOS Intel 64
- Delphi Android 64
- Delphi iOS 64
- C++ iOS 64
- Delphi Linux 64

Not used for:

- Delphi and C++ Android 32
- Delphi Windows 64
- Delphi and C++ Windows 32

Making great progress!

III. RAD 11.2 Libraries Quality & Improvements



11.2 RTL: Quality Focus

- ZLib updated to 1.1.12 to address recently patched vulnerability, plus other zip quality (stored files, symlinks on Linux, handling truncated data, and more) and performance
- New: TProxySubrangeStream class: a subrange of any other stream (similar to previous private TStoredStream)
- PCRE is JIT-enabled, giving great performance
 - and can access TPerlRegEx.Study
- RTTI quality
- System.SysUtils.StrToFloatDef no longer changes the numerical exception mask
- Several performance improvements

11.2 VCL: Quality Focus

- Improvements in the use of Microsoft's WebView 2 control in both TEdgeBrowser and TWebBrowser components
 - Better support for the UserDataFolder and the ExecutableFolder configurations
 - More flexibility in the deployment of applications that use the WebView 2 control, including secure enterprise scenarios
- Many issues addressed for CardPanel, NumberBox, RichEdit, ListView, DateTimePicker and BalloonHint components

11.2 FMX: Quality & Performance Focus

- Support for iOS status bar with dark icons
- TPathData quality
 - Extensive improvements over TPathData processing and rendering, both in terms of quality and performance, including quadratic Bézier curve commands and other transformations
 - Increased the rendering speed of TPath on GPU canvas
- Many other quality improvements, including areas:
 - TBufferedLayout reacts to an update of contained control styles
 - Windows privacy and TVideoCaptureDevice interact well, plus other Windows camera quality
 - TWebBrowser and Edge quality
 - StyleBook Windows 64 quality
 - ListView quality with DynamicAppearance
- TListBox supports Shift+Click to select a range

11.2 Data

- Search and Replace with FireDAC reFind
 - Script for dbGo migration
 - plus compatibility unit, FireDAC.ADO.Migrate.pas
 - reFind also lets you leave old code as a comment (/C switch)
- Updated FireDAC drivers
 - ODBC for DSQL Server
 - MSSQL on macOS M1
 - support for MongoDB new transaction feature
- Addtl DB library changes



11.2 Data

- FDMonitor greatly improved!
 - Syntax highlighting improvements
 - SQL Output view to track SQL commands
 - FDExplorer and FDMonitor have search commands
 - Optimizations of ODBC and other
- FireDAC SQL editor has syntax highlighting, line numbers, and common shortcut keys
- TCustomADODataset.AutoIncFieldsMode: optimise opening a dataset by avoiding extra data traffic to detect auto-increment fields



11.2 RAD Server and HTTP/REST

- Enhanced Logging: now log IP address and DateTime
- emsserver.ini can specify paths to the FDConnectionDefs/FDDrivers.ini files
- EMSFireDACClient component has a context menu item to create persistent fields based on result set
- TEMSFireDACClient.PostUpdates can use a custom session token
- TEMSFileTResource list function improved

HTTP and REST

- Improved Windows proxy detection
- Improved THHTTPNetClient SSL certificate checks for Android
- TRestRequest MIME types and multipart/form data improved



11.2 Quality Portal and Overall Quality

- Specific fixes for over 450 QP issues (1070+ in 11.x)
 - New Features: 30
 - Bugs: 420*

* Estimate, pending final testing on a few issues, doesn't include duplicates and 'won't fix'/'works as expected' items

- QP issues addressed by product area:
 - IDE: over 350 issues
 - FireMonkey and VCL: around 100 issues each
 - RTL, Data, Compilers: over 75 issues for each area



Additional Product Initiatives

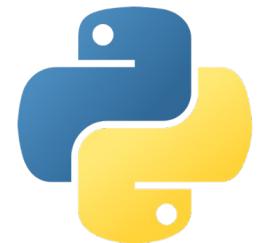
- Appercept's AWS SDK for Delphi
 - Long term distribution deal, included in Enterprise and Architect
 - New releases expected at a constant pace



Python

Based on Python4Delphi

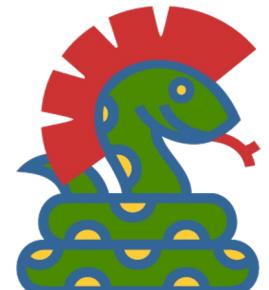
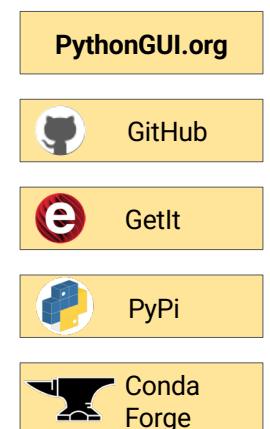
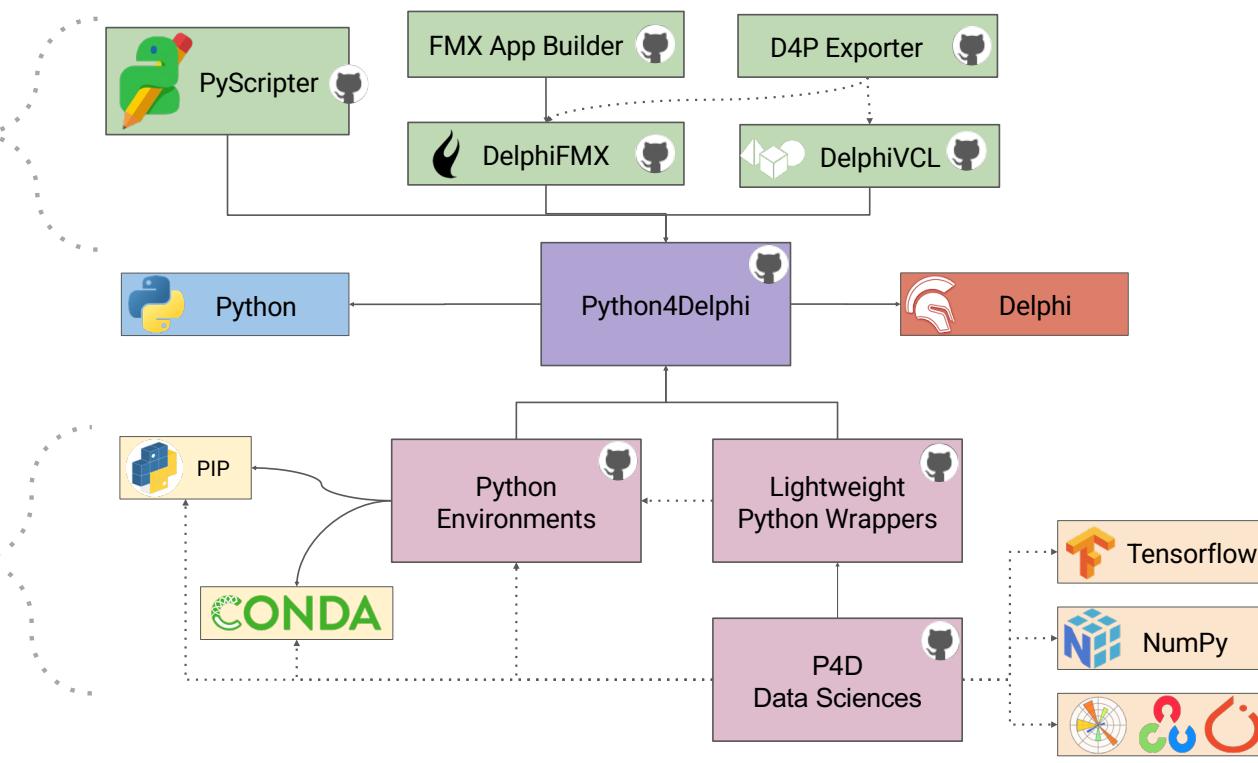
- Delphi UI libraries (VCL + FMX) for Python
 - Free UI library for Python developers
 - Upgrade to a Delphi license for additional power and features
- Also, you can use Python in Delphi applications



Embarcadero Python Ecosystem

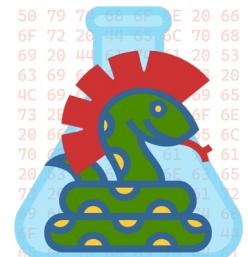
For Python Developers

For Delphi Developers



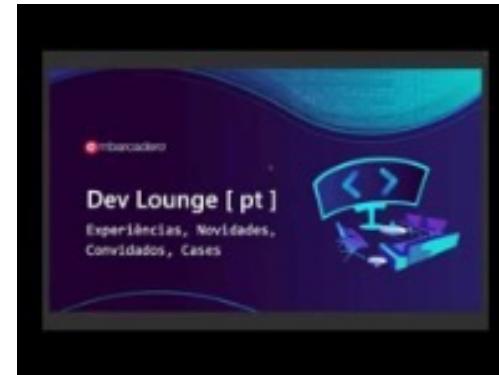
Embarcadero Python Ecosystem

- **Python4Delphi:** The Bridge between Delphi and Python
- **Python Environments:** Simplifies deployment and configuration of Python
- **Lightweight Python Wrappers:** Wrap Python libraries into Delphi components
- **P4D Data Sciences:** Delphi components for popular Python Data Sciences tools
 - Tensorflow, NumPy, PyTorch Vision, PyTorch, Matplotlib, Natural Language Toolkit (NLTK), OpenCV, and Scikit-learn and more
- **DelphiVCL for Python:** VCL as a Python library for native Windows GUI
- **DelphiFMX for Python:** FMX as Python library for cross platform GUI
- **D4P Exporter:** Export FMX & VCL forms for use in Python
- **FMX App Builder:** Build & debug pure Python powered Android apps
- *And more coming in the future!*

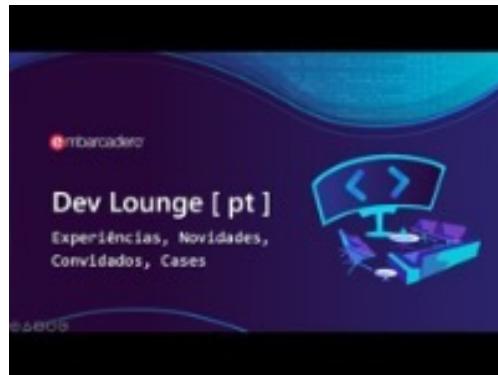




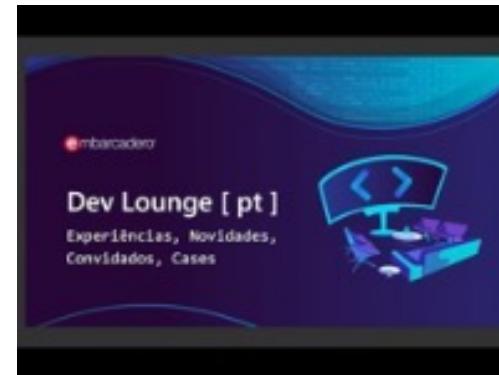
[Dev Lounge - Episódio #6
\(Azure\)](#)



[Dev Lounge - Episódio #7
\(AWS\)](#)



[Dev Lounge - Episódio #8
\(Skia4Delphi\)](#)



[Dev Lounge - Episódio #10
\(Delphi e Python\)](#)

RAD Studio 11.2: iOS Simulator & Quality / Polish

I. Focus on Quality and Improvements

- IDE and Designer High DPI
- Usability / 'The Small Things' Quality

II. IDE Improvements

- CPU View Syntax Highlighting
- Markdown Support
- Editor Tab Improvements
- Fonts and Styles

III. Code Insight / DelphiLSP

- Reworked Help Insight (and using XSLT again)
- Marking Inactive Code and Color Constants
- C++ Code Insight from 11.1.5

IV. Platforms and Tooling

- iOS Simulator for Delphi
- LLDB for Linux
- Android Target API 32
- Remote Debugging Win64 C++

V. Libraries

- VCL WebView2 Improvements
- General VCL and FMX Quality
- ZLib, HTTP / REST
- RAD Server
- FireDAC reFind, Updated Drivers, and FDMonitor/FDExplorer



Additional Resources

- <https://github.com/corneliusdavid/AutoGetIt>
- <https://www.markdownguide.org/cheat-sheet/>
- https://docwiki.embarcadero.com/RADStudio/Alexandria/en/XML_Documentation_Comments
- https://docwiki.embarcadero.com/RADStudio/Alexandria/en/Help_Insight
- https://docwiki.embarcadero.com/RADStudio/Alexandria/en/XML_Documentation_for_Delphi_Code
- https://docwiki.embarcadero.com/RADStudio/Alexandria/en/Compiler_XML_Output_Global_Declarations
- https://docwiki.embarcadero.com/RADStudio/Alexandria/en/Generating_Project_Documentation
- https://docwiki.embarcadero.com/RADStudio/Alexandria/en>Welcome_Page



Q & A

fernando.rizzato@embarcadero.com