

Introducing

RAD Studio 12



embarcadero®

Embarcadero and Idera

Embarcadero is owned by Idera, www.ideracorp.com

Part of the DevOps division along with other tools for developers

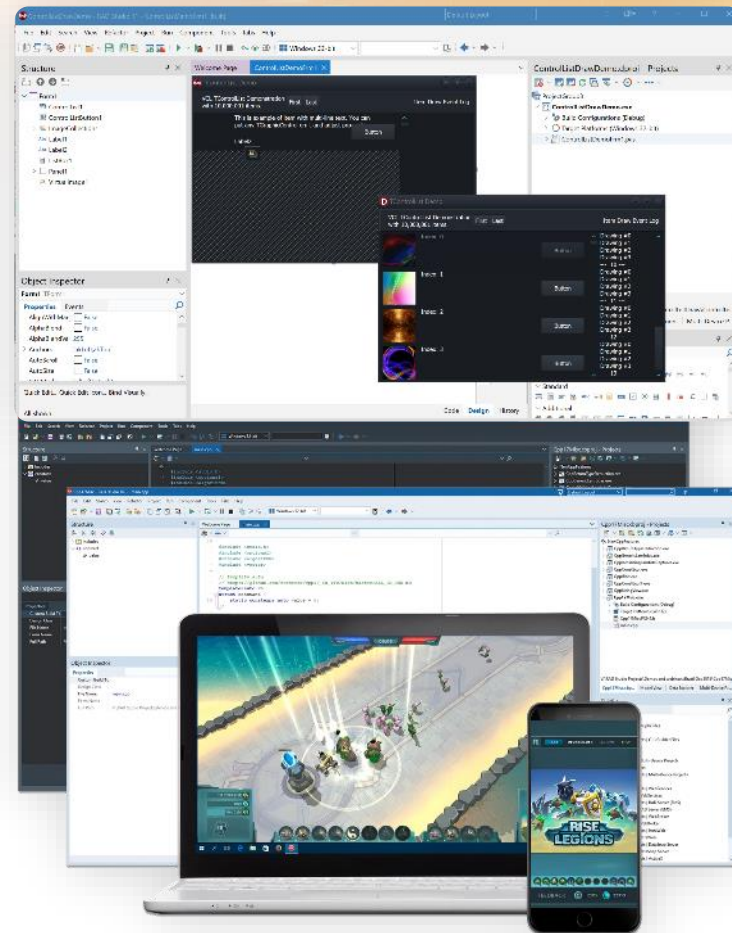


What is RAD Studio?

The ultimate IDE for building

- multi-platform
- high-performance
- native applications

in Modern Object Pascal
and Extended C++
with powerful visual design tools
and integrated toolchains



What is Great in RAD Studio?



Developer Productivity

Shipping is a feature. Get to market 5x faster.



Fast Native Apps

Native compilers give your apps the speed they need.



Strong Community

Technology Partners, MVPs, trainers, authors and developers – worldwide

Visual Designers
Forget prototyping in another tool, just design it and finish it in one IDE



Platform API Access

Gives you access to all the platform APIs on all platforms.

Database Access

Key to Delphi's initial design, database access is easy and fast.



Backward Compatibility

You have an investment in your code, keep your code relevant.



RAD Studio 12 **Athens**

RAD Studio 12 Athens

- Includes:
 - C++ Builder 12 Athens with Visual Assist for C++ Builder
 - Delphi 12 Athens



Agenda: A Mega Release For You Today!

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?!*

Live Q&A! That was a mega webinar...

- ***Now let's answer all your questions!***



A New C++ Builder Experience



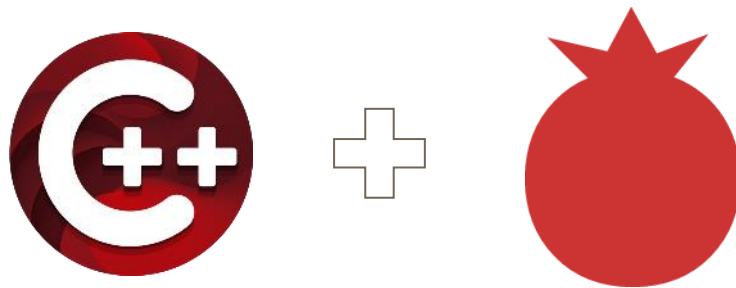
Visual Assist for C++ Builder

**One of the most amazing things
we've ever done for RAD Studio**

Code Insight, Refactorings, & Navigations

Visual Assist for C++ Builder

A top-notch set of coding and productivity tooling from the popular Visual Studio tool, by sister company Whole Tomato



**Massive productivity boost:
Code Insight, Refactorings, Navigation**

1. Code Completion

Fast code completion: very speedy to display

Completely different engine

Understands types, but also *heuristic* in results

```
#include <iostream>
#include <tchar.h>

int _tmain(int argc, _TCHAR* argv[])
{
    std::map
```

- class hash_map
- class hash_multimap
- class map
- class multimap
- class unordered_map
- class unordered_multimap

```
PaintGrid();
Ac
```

- property Action
- function ActionChange
- variable ActionLink
- function Activate
- property Active
- function ActiveChanged

- __property Action
- __property Action

Even suggests items not #include-d,
from headers VA knows about!

Plus: tooltips, parameter completion...

2. Refactorings

You have code and want to:

- Rename a method or type, safely, everywhere it's used in your project group
- You wrote code and now it won't compile and you have to scroll to the top to add a `#include`. If only this could be done for you...
- You wrote a method and want the declaration automatically added – or vice versa

Visual Assist does these!

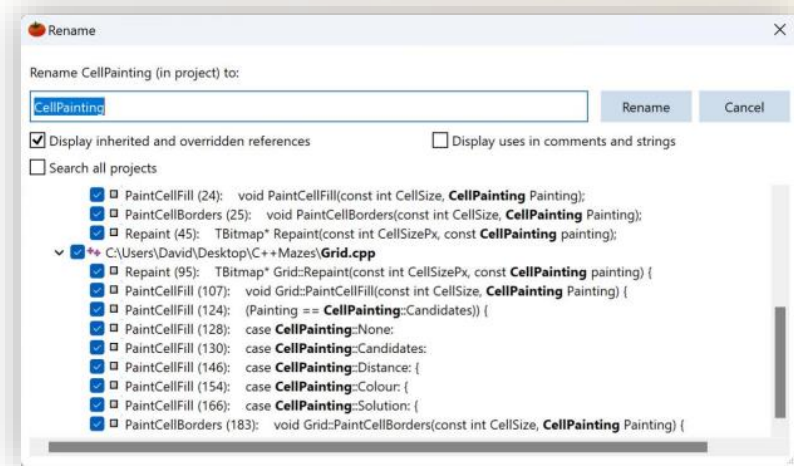
2. Refactorings: *Rename, Add Include, & more*

Rename symbols quickly and accurately, in your project or throughout your project group.

Add the right #include file for any symbol:



Create the implementation or declaration of any method



Let's see a demo!

3. Navigations

You want to move around your code!

- Move to a type or a method definition
- See what uses a method or class
- Navigate to a symbol implementation
- See the ancestors of a type
- List the descendant methods override of a virtual method

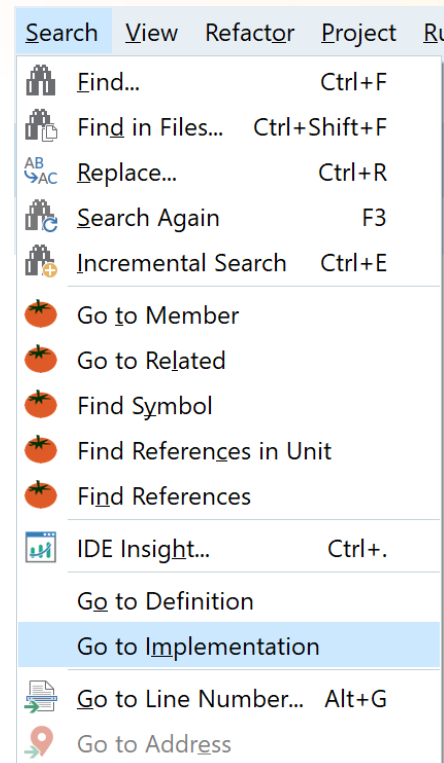
Visual Assist does these!

3. Navigation: *Toggle between impl & decl*

- Go To Implementation simply goes where the method is implemented
- Go To Declaration goes to where the method is defined

The same Ctrl+Shift+Up/Down keyboard shortcut as Delphi to toggle between the two

One of our most requested C++ features!

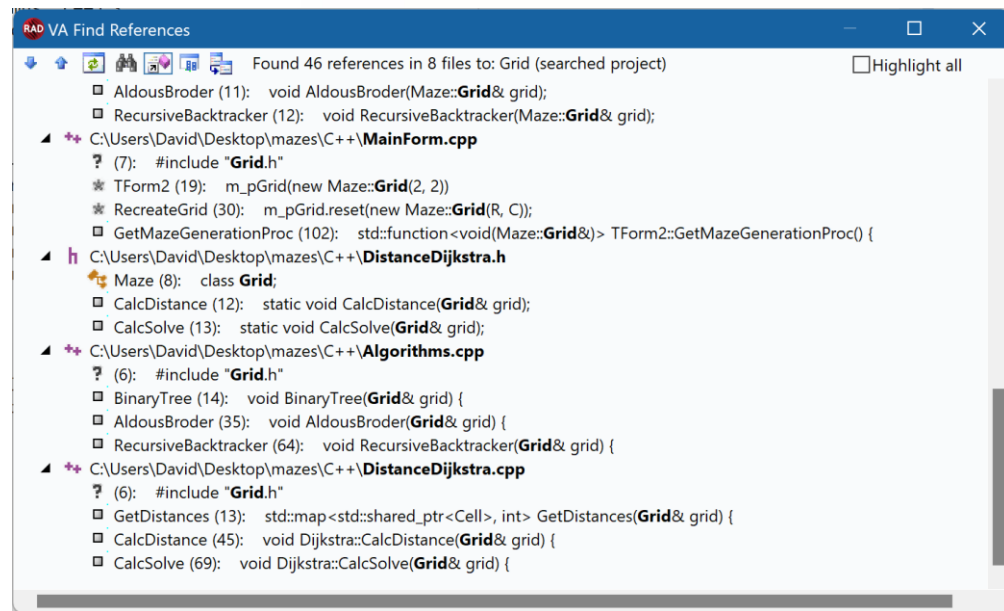


3. Navigation: *Find References*

Find references to any symbol (method, type etc.) Where is it used?

Not a search. Aware of your code: this is the result of a database query and reflects usage not just matching names

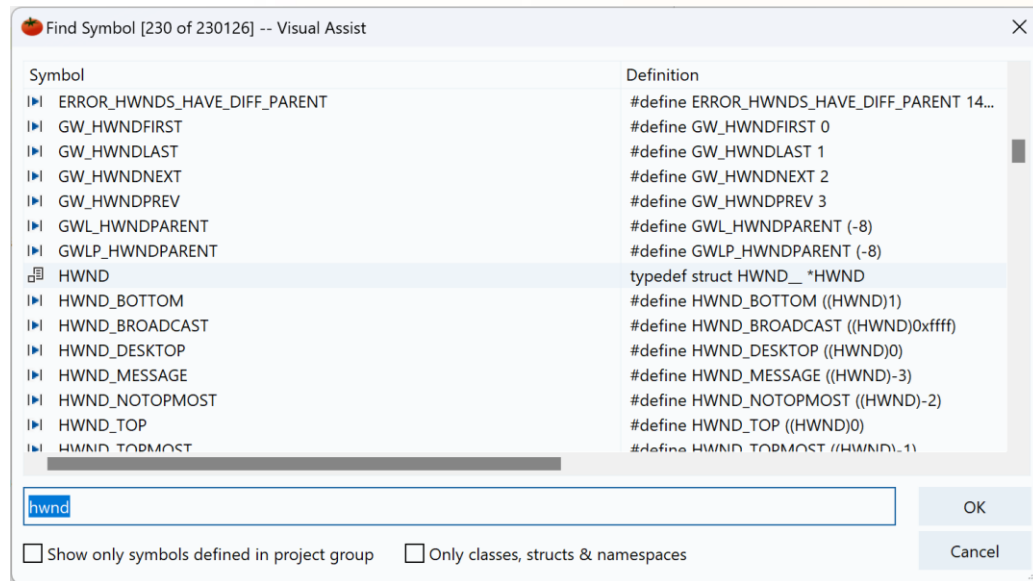
Find in project / project group
Addtl menu for quick find in unit



3. Navigation: *Find Symbol*

Find symbols in your project or project group

- Very fast: database queries not searches
- Powerful search: can filter substrings, begins-with, ends-with, exclude
- Aware of the headers your project uses



Let's see a demo!

3. Navigation: *Go To Related*

The single most amazing feature ever.

```
TBitman()
```

Definition FMX.Graphics.hpp:1648

Declaration FMX.Graphics.hpp:67

Base Classes ▶

Derived Classes ▶

Constructors ▶

Goto Member...

```
();
```

Control-click on any symbol

Works on methods, variables, parameters, types (classes, structs, enums...), include files

Go to:

- Definition
- Declaration
- Constructor
- Ancestor or descendant classes
- Ancestor or descendant virtual method overrides
- Any member

Visual Assist in C++ Builder

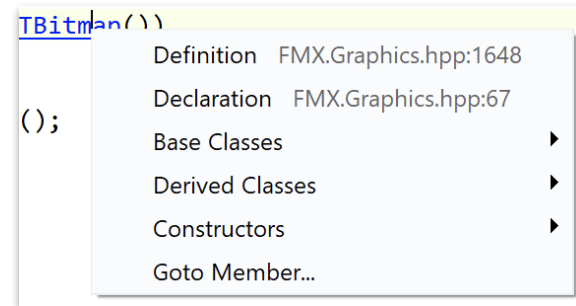
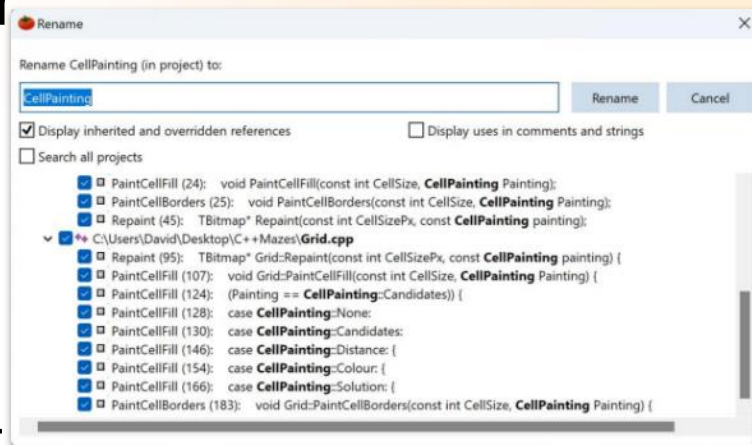
An amazing initial integration


- Code completion: fast, heuristic
- Refactoring: Rename, Add Include, and create implementation/declaration
- Navigations: Find References, Find Symbol, toggle impl/decl, Go To Member
- And Go To Related: the single most amazing feature ever

Huge for productivity and daily workflow

Fast, speedy, contextual.

Code Insight (completion etc) uses VA by default





New Win64 Clang Toolchain Preview

Preview: New Win64 Clang compiler

- Building a new C++ Win64 compiler based on a recent version of Clang
- Support for the most recent C++ language features and a much more stable compiler and toolchain
- A new foundation for the future of our C++ compilers
- In 12.0 we are shipping a Preview of this new C++ toolchain for Win64, available as a command line compiler

[https://blogs.embarcadero.com/
win64-clang-toolchains-in-rad-studio-12/](https://blogs.embarcadero.com/win64-clang-toolchains-in-rad-studio-12/)



[Yukon Beta Blog]: Win64 Clang
Toolchains in RAD Studio 12

C++ Clang 15 Compiler Goals

- Modern C++ standards
 - Run C++ code you find online, use third party libraries
 - Use modern safe coding standards
 - Write code you expect to work and have it work
- Performant apps
- Excellent debugging
- Excellent toolchain tools, eg the linker
- Matches platform standards (eg COFF and PDB)
- Supports existing code with C++Builder-isms and links with RAD Studio Delphi libraries (of course!)

The C++ Tech Behind the new Compiler

Clang 15 - *modern toolchain*

Our extensions - *key for C++Builder!*

Win64 - *modern, key*

COFF object format - *platform standards*

PDB debug format - *platform standards, good debugging*

LLVM lld linker - *quality*

C RTL: UCRT - *platform-provided, quality, performance*

C++ RTL: homebrew, LLVM, Mingw - *quality, modern*

STL: LLVM's libc++ - *quality, performance, modern C++*

Preview: New Win64 Clang compiler

In RAD Studio 12, a technical preview

Demonstrates:

- Command line building, and IDE debugging, in this preview
- Platform standards: COFF and PDB
- New STL, and the quality that will come with it
- New linker, and the quality that will come with it
- (The rest: new C / C++ / etc runtime and more)

Not ready for a VCL/FMX app yet, but test out C++. Test it out!

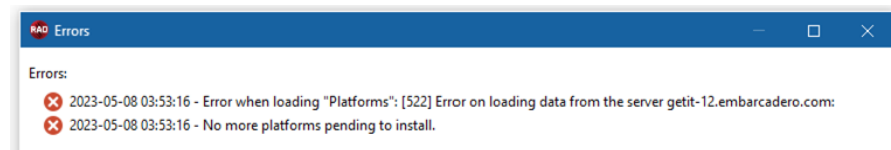
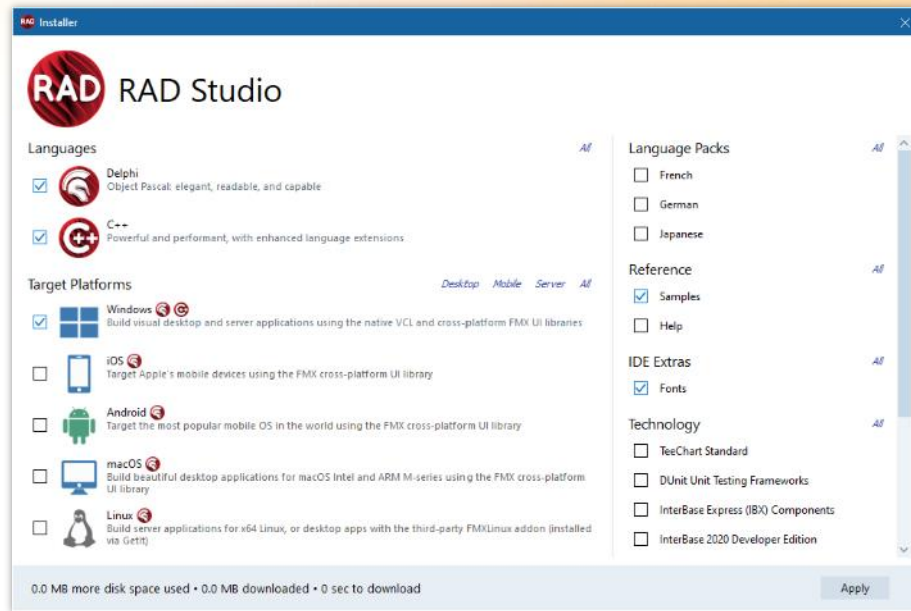


Installation and IDE

New installer UX

Significantly reworked and simplified installer: much easier to choose what will be installed

- Simplify installation choices.
- Everything is on one page: Choose languages, platforms, extras – done.
- Better info on the effect of changes
- Improved error messages, easier access to logs
- Uses VCL styles, supports high DPI



Syntax Highlighting throughout the IDE

Highlighting added to

- Call Stack
- Debugger tooltips
- Error Insight messages
- Structure view
- Navigation toolbar

Watch List - Thread 9628

Watch Name	Value
<input checked="" type="checkbox"/> Caption	'Form3'
<input checked="" type="checkbox"/> ClientRect	(0, 0, 1248, 882, (0, 0), (1248, 882))
<input checked="" type="checkbox"/> @Self	\$9BF800

Call Stack - Thread 4304

→ Unit3.TForm3.**FormCreate**(\$3382820)

- Vcl.Forms.TCustomForm.**DoCreate**
- Vcl.Forms.TCustomForm.**AfterConstruction**
- System.**_AfterConstruction**(\$3382820)
- Vcl.Forms.TCustomForm.**Create**(\$33ACD10)
- Vcl.Forms.TApplication.**CreateForm**(TForm3,(no value))
- Project3.**Project3**
- :75027ba9 KERNEL32.BaseThreadInitThunk + 0x19
- :76fab79b ntdll.RtlInitializeExceptionChain + 0x6b
- :76fab71f ntdll.RtlClearBits + 0xbfb

Structure

GetImageIndex: Integer

GetTabHintText: string

Close(var Allowed: Boolean)

GetTabColor: TColor

LoadViewState(const Desktop: TCustomIniFile; const ViewDeskSection: string)

AllocCoTaskMemStr(const S: string): LPCWSTR

ApplicationMainHandle: HWND

CenterWindow(Wnd: HWND)

CopyData(Handle: THandle): THandle

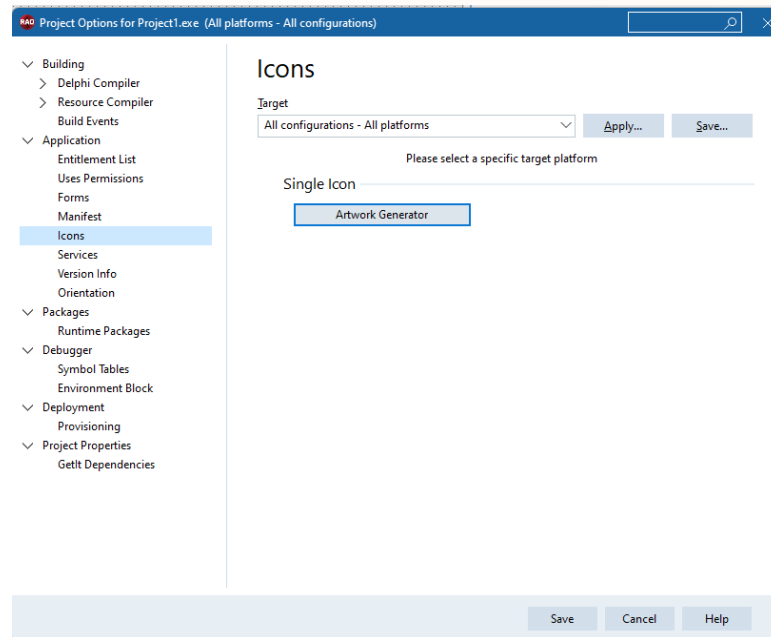
CreateMessageDialog(const Msg: string; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons): TForm

CreateMessageDialog(const Msg: string; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons): TForm

< 337 result(s) found >

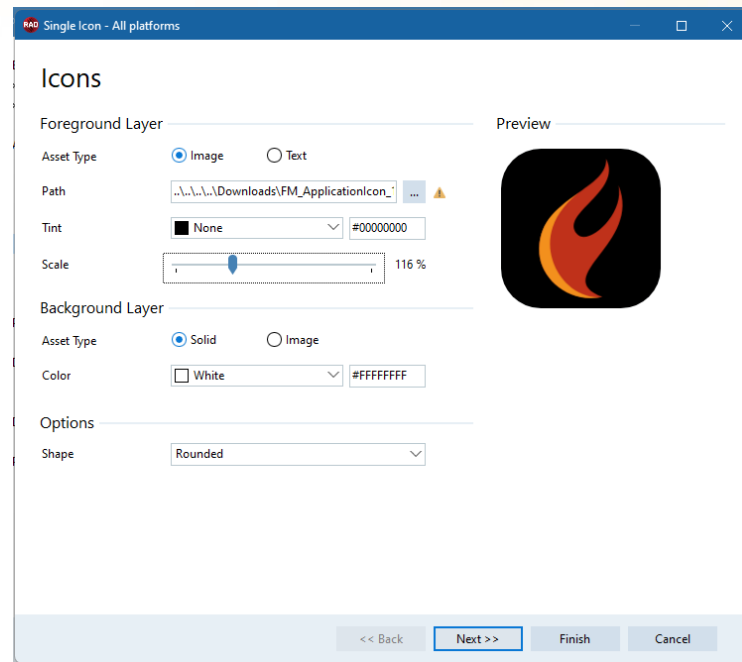
Multi Device Icon Generator

- Generates icons for all screen resolutions
- Generates splash screens
- Generates Android's adaptive icon
- Embedded in the IDE as a multi step wizard



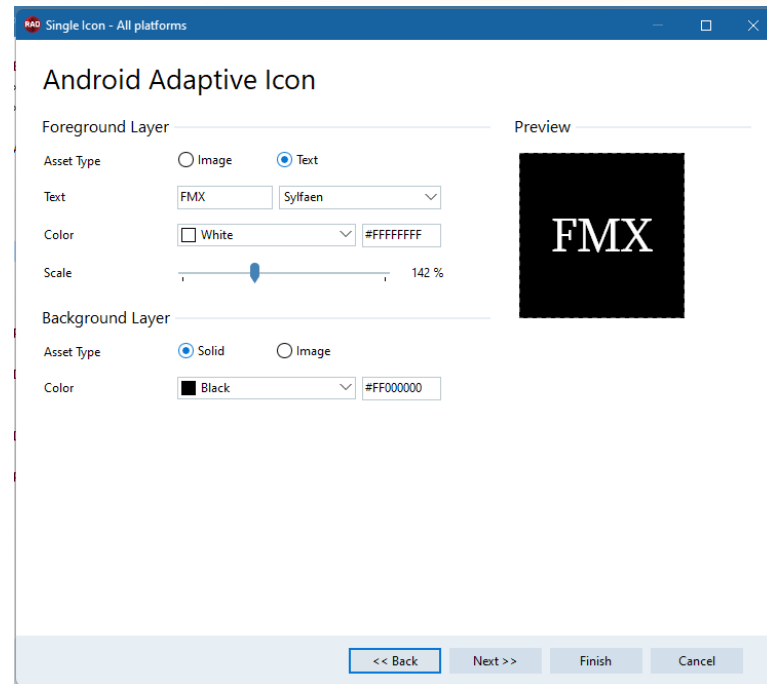
Icon Configuration

- Can be created based on a image or text
- Color and scale can be adjusted
- Icon can be rounded, square...



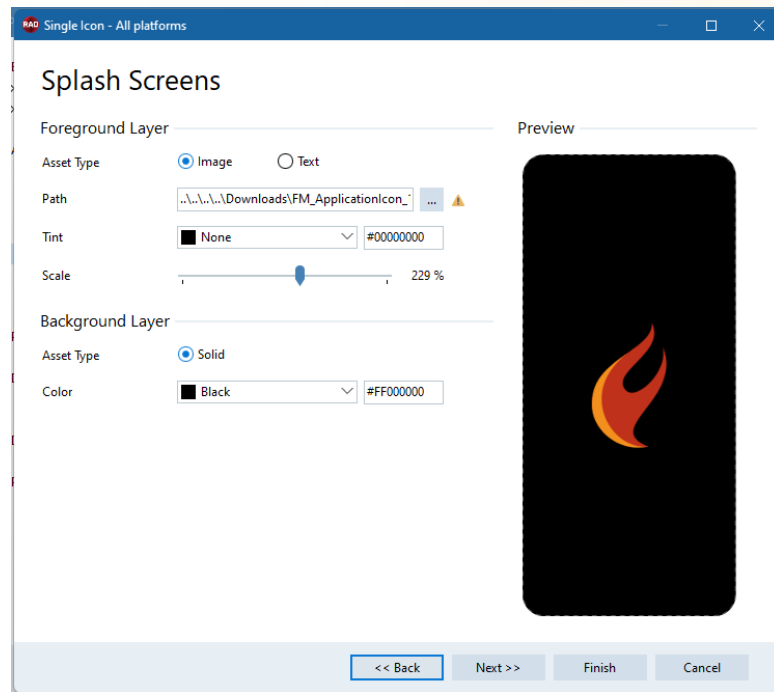
Android Adaptive Icon

- Similar options to a standard icon
- If you want to use this option with an image, you will need to use a SVG



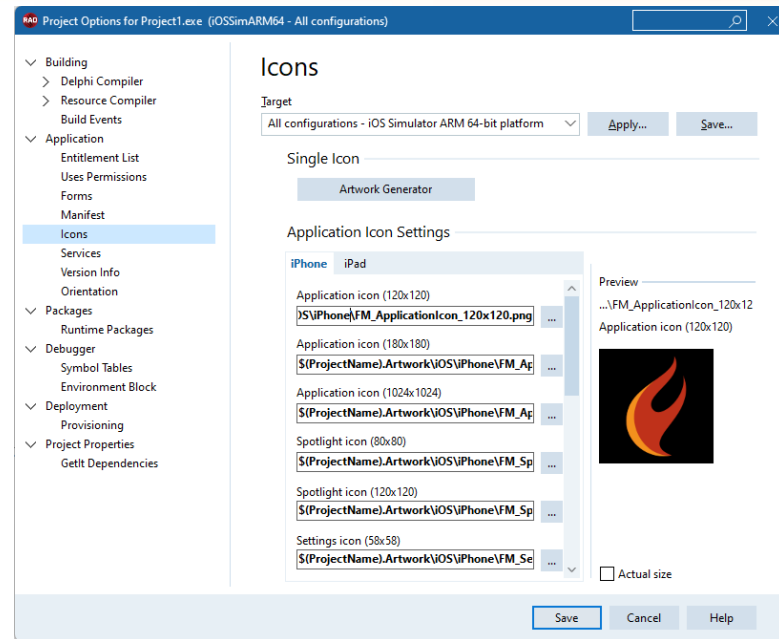
Splash Screens

- Based on your upload image
- Generate splash screens for light and dark theme



Multi Device Icon Generator

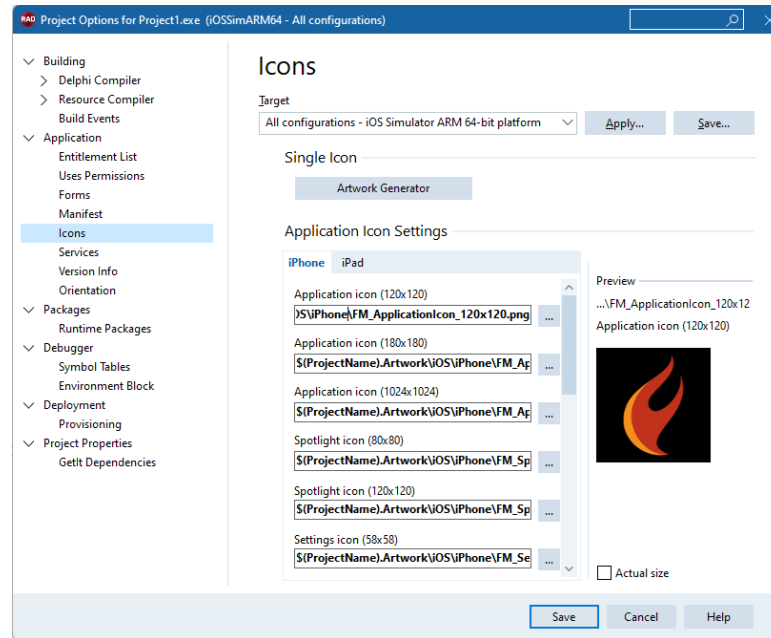
- All icons can be generated at once or individually
- Changes to single icons for specific platforms can be made easily



Multi Device Icon Generator

- All icons can be generated at once or individually
- Changes to single icons for specific platforms can be made easily

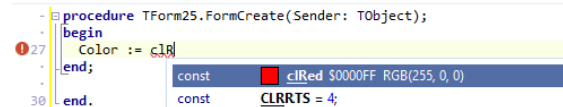
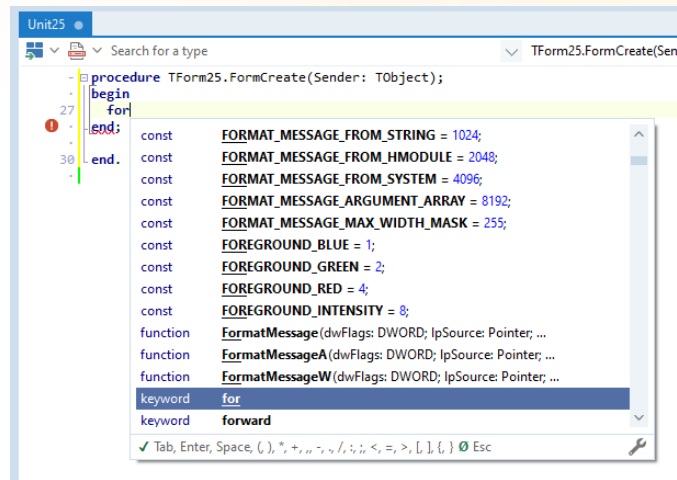
Keep in mind that the generated images are saved in a sub-folder of the project folder with the project name. Therefore, if you rename the project, you must manually rename the subfolder accordingly.



Delphi LSP

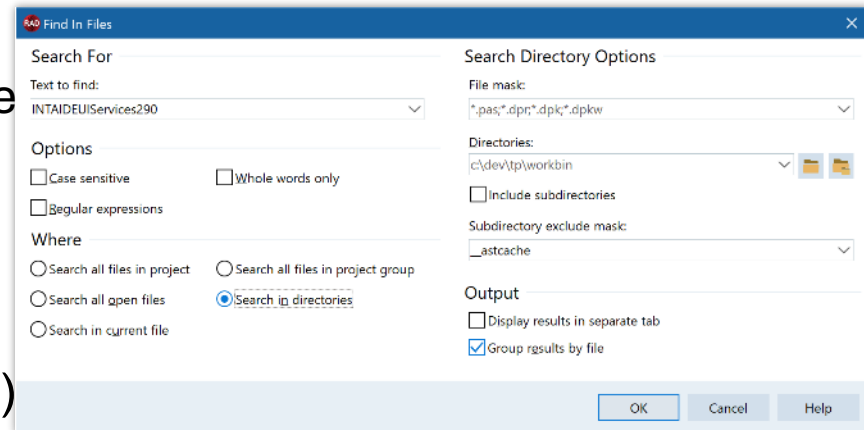
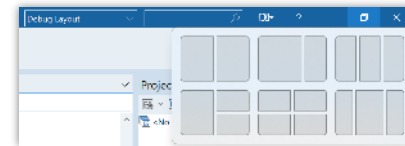
Improvements focused on stability, especially closing or switching projects. Plus:

- Autocompleting [] for array types
- Improved color constant representation
- Re-introduced code completion's auto-invoke feature (off by default in this release)
- Language keywords included in the completion list
- Templates still included, but in correct locations
- New menu item to restart the LSP server
- The code completion window can show the keys that affect completion



IDE: Smaller Features

- New Win32/WinAPI App Wizard for C++
- You can reopen a tab with the state (scroll pos, cursor pos, region collapsing, bookmarks, etc) exactly as it was (via a setting on the Options > IDE > Saving and Recovering page)
- Editor tabs now have a Copy File Path menu item, for full / filename only / path only
- The VCL designer will now show custom colors set for controls even when using the 'Mimic System Style' setting
- The IDE now supports Windows 11 Snap Layouts in the title bar
- Searching lets you exclude folders (with defaults set up for C++ and Delphi)



IDE Quality

Lots of quality work, but especially want to note around:

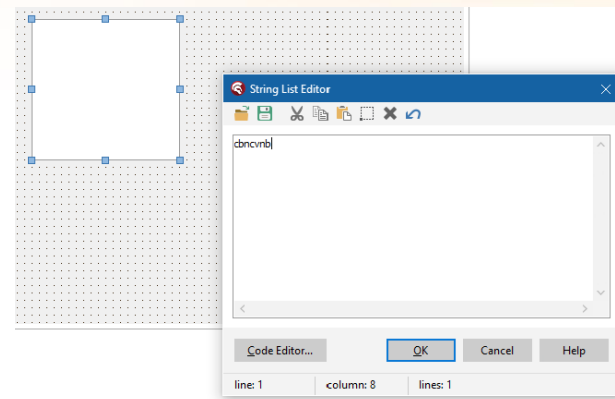
- Multiple editor windows
- Searching (eg unwanted behaviour for editor double clicks, or whole word searches, etc)
- The VCL form designer including in high DPI mode

GDI resource usage should be radically decreased, eg from ~4000 to ~1500 with several component packs installed; we make the ability to implement the same thing available in the VCL

New VCL Designers (from Ksvc)

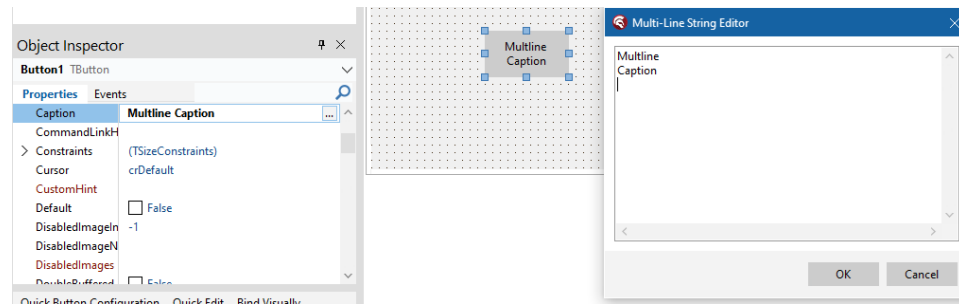
New StringList Editor

- Toolbar to load and save to text files, standard clipboard operations, plus a status bar with position information



Multiline String Editor

- Offering the ability to display multiple lines of text in their Caption (or other) properties



New VCL Quick Editors (from KSVC)

Similar to other “Quick” editors already in the IDE

Quick RadioGroup

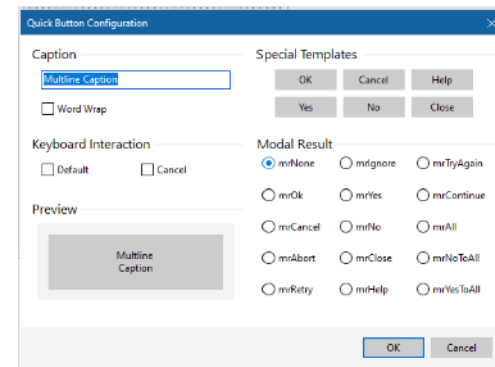
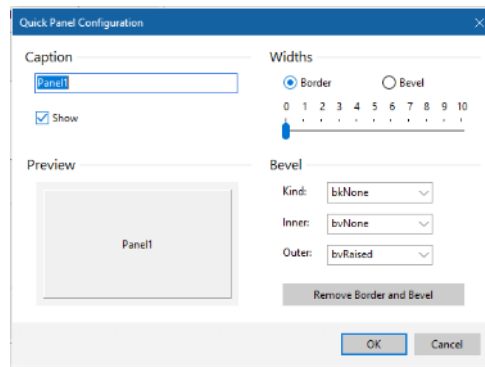
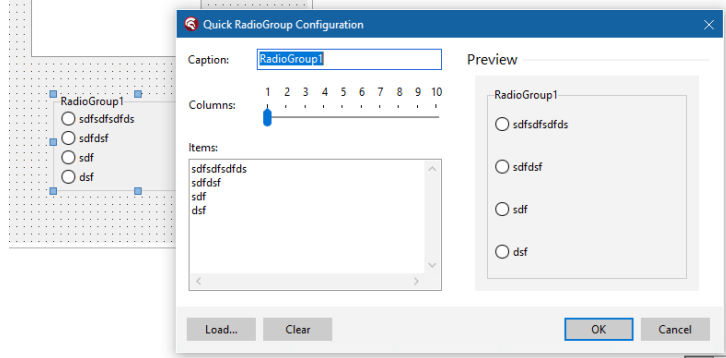
Quick Panel

Quick Button Configuration

Configuration

Configuration

Configuration



Debuggers

The new C++ Win64 platform in Preview uses LLDB 15 and PDB debug format, making it much more accessible to other tooling

Other changes:

- PAServer messages show in the Messages pane
- OS thread handle shows in the Threads view
- 'Debugger Not Responding' improvements: used to be a single long timeout; now much shorter, but grows each time you press the new Wait button

ToolsAPI

We love the ToolsAPI: it lets our amazing developers build amazing things!

New APIs building on last release's editor info and editor painting APIs:

- Line state (eg, 'inactive breakpoint', 'compiled with blue dot', etc)
- Cell state (eg, selected, hotlinkable, search match, syncedit, brace match, etc)

Should assist fully custom painting.

New APIs for reading and writing to the editor

- Much higher performance than existing stream/file-like APIs; random access supported

Misc improvements: assist setting up IDE title bar, getting editor token, themed input and message dialogs (same as IDE uses), plus some quality



UI Libraries: FireMonkey

What's New in FireMonkey? Really a lot!

- Skia Graphic Library library support
 - For all target platform, both Delphi and C++Builder
- Completed the redesign of styled TEdit/TMemo
- Android API 33 Support (see later)
- New Icon and Splash screen designer (covered earlier)
- Split Views for mobile platforms
- Plus smaller features and quality

Skia and FireMonkey

Why Skia?

Skia is a multi-platform 2D graphics library

Skia offers high performance and high quality rendering

Skia supports advanced graphic operations across all platforms

Skia offers support for many image and video formats

Skia is becoming the new foundation for FireMonkey rendering

What is Skia?

“Skia is an open source 2D graphics library which provides common APIs that work across a variety of hardware and software platforms. It serves as the graphics engine for Google Chrome and ChromeOS, Android, Flutter, and many other products.” - skia.org



Skia logo source: https://en.wikipedia.org/wiki/Skia_Graphics_Engine

How we leverage Skia?

“Skia4Delphi is a cross-platform 2D graphics API for Delphi and C++ Builder based on Google's Skia Graphics Library. Provides common 2D APIs by abstracting complexities in implementing low-level libraries used behind, such as OpenGL, Vulkan, DirectX, and Metal, among others”



Main Features of Skia (1/2)

Feature	Details
2D drawing	Shapes, paths, and texts
SVG	Render and creation
Image decoders	BMP, GIF, ICO, JPG, PNG, WBMP, WEBP, and raw images
Image encoders	PNG, JPG, and WEBP
Animations player	Lottie, Telegram Stickers, animated GIFs, and animated WEBP
Anti-aliasing	High draw quality, no jagged edges
Font	Font weight, families fallbacks, and custom font (without installation), ligature

Main Features of Skia (2/2)

Feature	Details
Text	Multiple styles, max lines, line spacing, justified text, text outline, gradient, and decorations
Right-To-Left languages	Rendering of texts in Persian, Arabic, Hebrew, etc
PDF	Generation of vectorized PDF
Unicode	Graphemes parser
Filters	Color, mask, and image filters
Clippings	Support for many advanced clipping operations such as paths and shaders
Gradients	Linear, radial, sweep, and conical gradients
Shader	Creation of shaders to execute specific draws directly on the GPU, through a single shader language (SkSL)

Skia in RAD Studio

Different levels of usage for both Delphi and C++Builder

1. Skia API

Access to the pure Skia library, through a single unit: Skia.pas or Skia.hpp

```
Unit25 System.Skia
Search for a type Search for a method

{ TSkPictureRecorder }
TSkPictureRecorder = class(TSkObject, ISkPictureRecorder)
strict private
function BeginRecording(const AWidth, AHeight: Single): ISkCanvas; overload;
function BeginRecording(const ABounds: TRectF): ISkCanvas; overload;
function FinishRecording: ISkPicture; overload;
function FinishRecording(const ACullRect: TRectF): ISkPicture; overload;
2410 public
constructor Create;
class procedure __DestroyHandle(const AHandle: sk_handle_t); override;
end;

{$HPPERMIT END 'define SkPictureRecorder(...) __SkCreate(TSkPictureRecorder, ISkPictureRe

{ ISkPixmap }

ISkPixmap = interface(ISkObject)
2420 ['{FEFDBF69-4C8D-4124-83A6-68A981DF239E}']
function Erase(const AColor: TAlphaColor): Boolean; overload;
function Erase(const AColor: TAlphaColor; const ASubset: TRectF): Boolean; overload;
function Erase(const AColor: TAlphaColorF; AColorSpace: ISkColorSpace = nil): Boolean;
function Erase(const AColor: TAlphaColorF; const ASubset: TRectF; AColorSpace: ISkColor
function ExtractSubset(const ADest: ISkPixmap; const AArea: TRect): Boolean;
function GetAlpha(const AX, AY: Integer): Single;
function GetAlphaType: TSkAlphaType;
function GetColor(const AX, AY: Integer): TAlphaColor;
function GetColorF(const AX, AY: Integer): TAlphaColorF;
2430 function GetColorSpace: ISkColorSpace;
function GetColorType: TSkColorType;
function GetHeight: Integer;
function GetImageInfo: TSkImageInfo;
function GetPixelAddr(const AX, AY: Integer): Pointer;
```

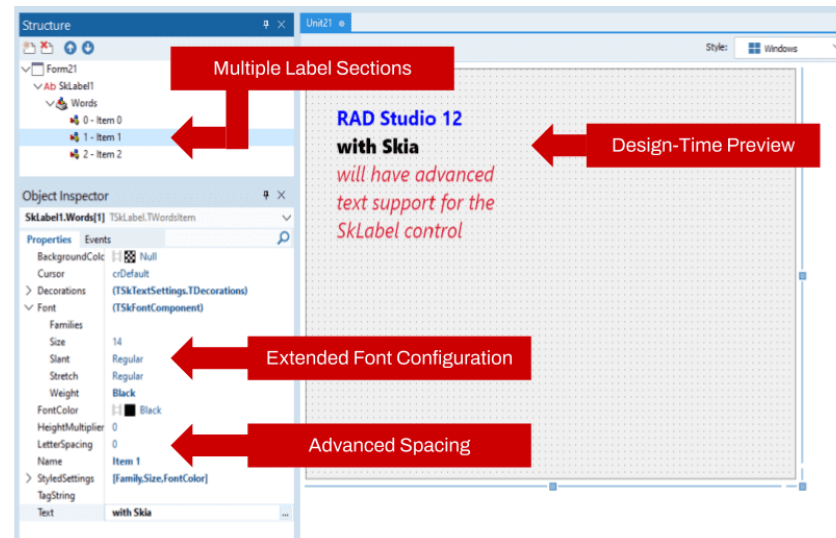
Skia in RAD Studio

Different levels of usage for both Delphi and C++Builder

2. UI Controls

(for both FMX and VCL)

- TSkAnimatedImage
- TSkLabel
- TSkPaintBox
- TSkSvg



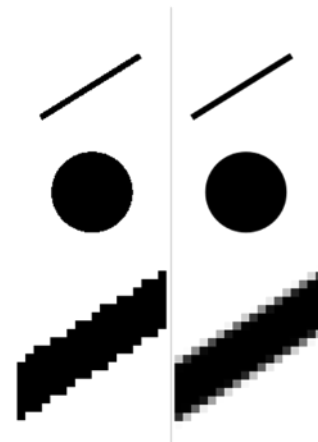
Skia in RAD Studio

Different levels of usage for both Delphi and C++ Builder

3. App and Styles Rendering

Replacement of FMX graphic engine with Skia. Better performance, improved anti-aliasing

FMX.Skia.GlobalUseSkia := True



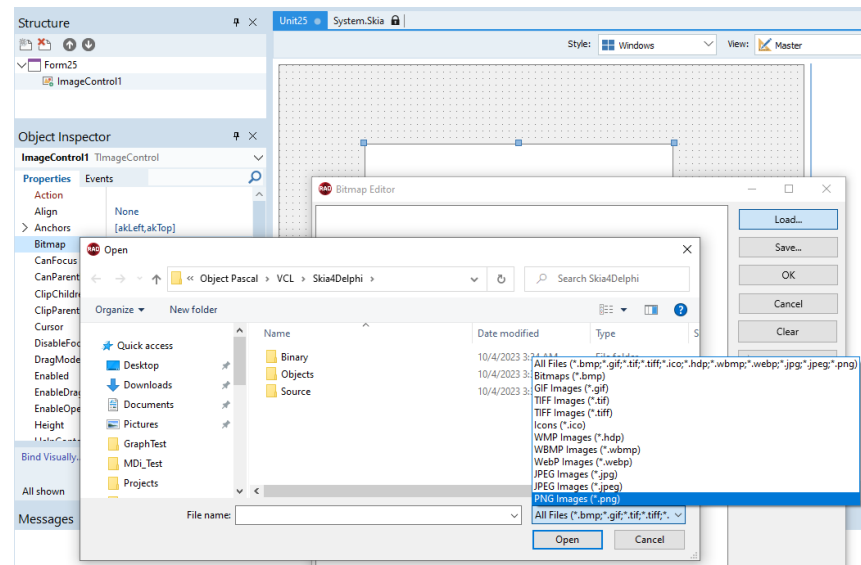
Improved drawing with anti-aliasing in Skia (on the right)

Skia in RAD Studio

Different levels of usage for both Delphi and C++Builder

4. Codecs for image controls

New image codecs Skia supports are registered in the frameworks, FMX or VCL image controls can directly load and save new image formats like WebP



New UI Controls (FMX/VCL) Based on Skia

- TSkAnimatedImage
 - Supports Lottie file, Telegram Sticker, Animated GIF, Animated WebP
- TSkLabel
 - Font weight, Font slant, multiple styles in text, BiDi (Right-to-Left)
 - Justify horizontal alignment, and much more
- TSkPaintBox & TSkAnimatedPaintBox
 - Paint with Skia APIs directly on the screen with the OnDraw event
- TSkSVG
 - Display SVG easily

FireMonkey Skia support for all platforms

For both in C++Builder and Delphi

FireMonkey, but also VCL

Leveraging the Skia4Delphi open-source project

Including additional capabilities not found in the open-source project:

- Vulkan support
 - Enhanced graphical performance and energy efficiency on Android compared to OpenGLES
- Skia Shading Language (SKSL) for effects and filters
- WebP Encoder
- Native printer for Windows and PDF printing for all platforms

Used also for new icon and splash wizard

FireMonkey: TEdit/TMemo Rework

Significantly refactored the TEdit/TMemo controls starting with desktop platforms (Windows and macOS) and also planning to include mobile ones

General FMX changes

- New TEdit property AutoSelect
- New TEdit CharCase
- Addressed several undo/redo issues using the recently added TUndoManager
- Addressed TEdit caret position problems, particularly on Windows
- TMemo implements IME Text inputting like in native TEdit
- TEdit takes into account the foreground brush resource
- Improved Shift-clicking selection

FireMonkey: TEdit/TMemo Rework

For the iOS platform

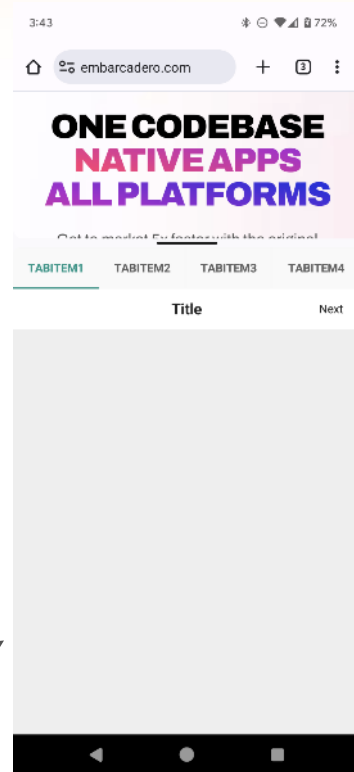
- Refactoring Virtual Keyboard: moved the logic for toolbar alignment/animation
- Improved the magnifier glass position for TMemo

For the Android platform

- Improved interactive text selection
- Adjusted selection Point position and place the left below text selection
- Added selection Point for position cursor in TMemo and TEdit
- Improved the PointInObjectLocal handles the area using a style with an offset of objects
- Updated platform and premium styles to support improved text selection
- Added CaretPoint for text-input controls for Android
- Support for finger slide gestures to move the cursor on TMemo on Android

Additional FireMonkey Improvements

- New TWinFormPositionerService to manage the screen(s) layout and positions
- Extensive refactoring of the TCommonCustomForm.KeyDown code, which handles accelerator keys, dialog keys, tab keys
- Changes to double click management
- New IFMXPlatformPropertiesService
 - Replacing IFMXDefaultPropertyValueService and IFMXTextEditingService
- Introduction of a new universal TFontManager
- New Windows 11 style
- Split Views / Split Screen support for Mobile: see image here





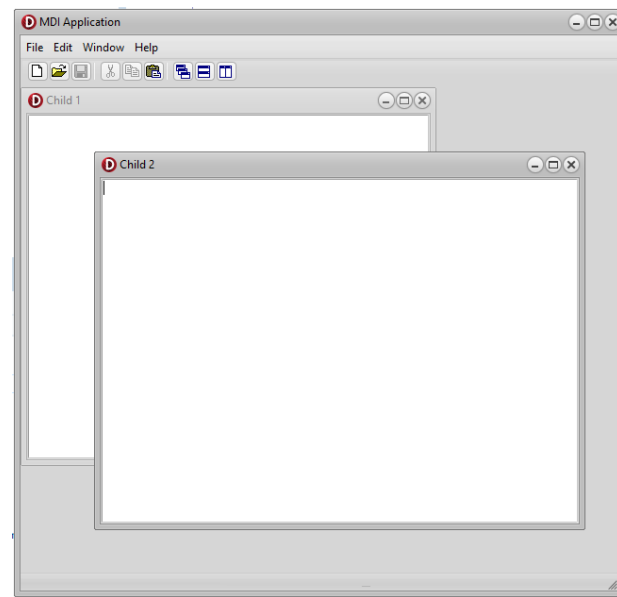
UI Libraries: VCL

Reworked VCL MDI architecture

Reworked the VCL MDI (Multi Document Interface) to address a number of platforms issues with HighDPI and VCL issues with styles:

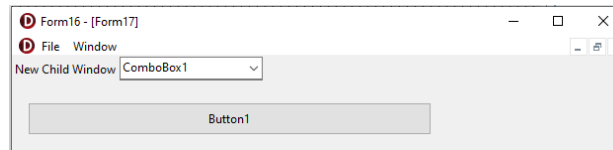
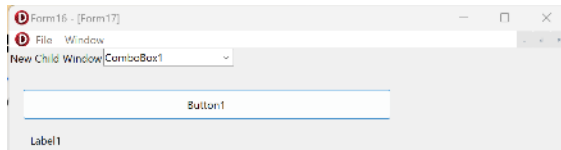
- Paint a custom border for MDI child window
- Support for a custom border for forms with Parent Assigned property (also outside of the MDI scenario)

Updated code generated by the MDI wizard (with support for tabbed UI)

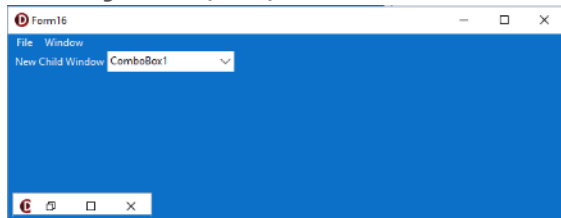


Reworked VCL MDI architecture (Examples)

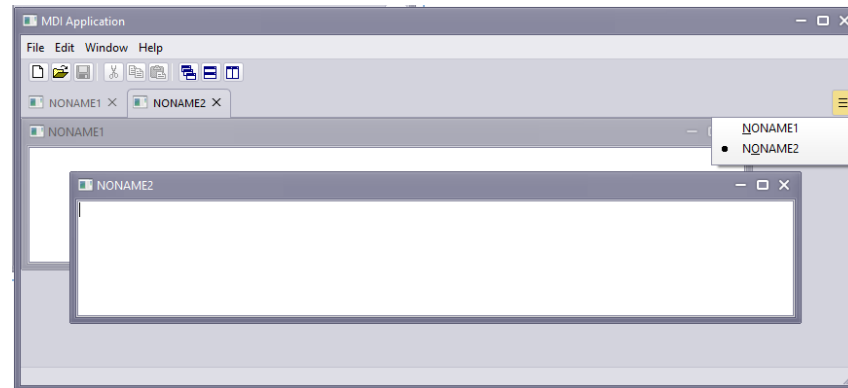
- Maximized child in unstyled app (11.3 -> 12)



- Minimized styles (12)

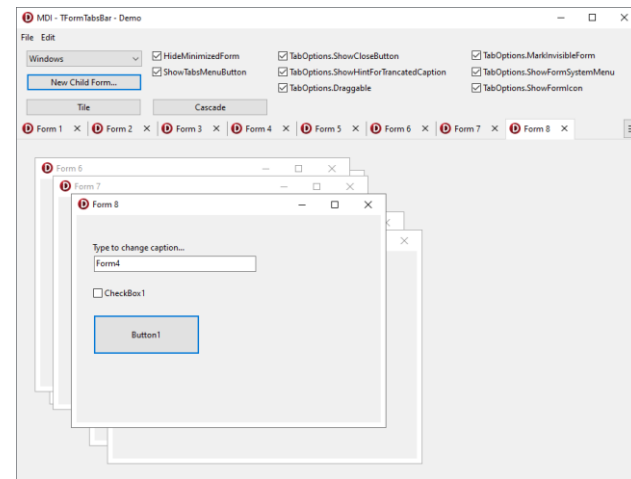


- App Generated by the MDI Wizard -> with styles applied



VCL: New Multi-tab architecture

- Goals
 - Migrate MDI apps to multi tab (full client area or partial, like in MDI)
 - Create new apps with a modern architecture out-of-the-box
- Introducing
 - IFormVisualManager Interface
 - VisualManager property of the TCustomForm class (to associate a form with its “*visual manager*” host)
 - TFormTabsBar Control
 - Ready-to-use solution for hosting multiple child forms in a modern tab-like user interface
 - Includes support for dragging a tab content out of the tab and in a separate window



VCL: Fonts and Screens

Reworked support for VCL TFont scaling independent from DPI scaling

- TFont.Size property now adapts to different DPIs
- TFont class new properties and methods:
 - property IsDPIRelated: Boolean // enables using TFont.PixelsPerInch property
 - property IsScreenFont: Boolean // for global fonts in TScreen class
 - procedure ChangeScale // called to initialize and scale any DPI-related font
 - procedure ScaleForDPI // used in code to adapt a font to any DPI

TControl Enumerator

Offers various filters to navigate child controls (which can be combined)

- `ceftAll` - all controls, including controls of the child controls (recursive)
- `ceftEnabled` - control should be enabled.
- `ceftDisabled` - control should be disabled.
- `ceftVisible` - control should be visible.
- `ceftInvisible` - control should be invisible.
- `ceftCustom` - Uses `CanEnumerateControl` method (`OnEnumerateControl` event)

```
for ACtrl in AControl.GetControls(AMyFilter) do  
begin  
    ...  
end;
```


Additional VCL Features (1/2)

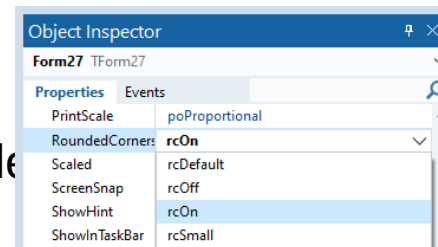
- Tile View support to TListView
 - New vsTile in TViewStyle
 - TileOptions property to adjust tile items
 - TileColumns property to adjust display of sub-item text from columns
- TNumberBox control
 - nbmInt64 mode accepting 64-bit numbers in input
 - Considers MinValue and MaxValue even if they are equal
- TControlList Control
 - Better support TControlList multiple selections
 - SmoothMouseWheelScrolling property
- The TitleBar control adds support for Windows 11 snap layouts

Additional VCL Features (2/2)

- TWICImage and TImageCollection now have a Dormant() method that removes GDI handles from system memory
- Significant rework of TActivityIndicator: new types (RotatingLines and Refresh) bitmap sizes, and custom colors



- ShowInTaskbar property for TForm (offering more flexibility in taskbar management)
- Desktop Windows Manager Enhancements
 - Win11 RoundedCorners and EnableImmersiveDarkMode
- New Double Buffering Mode



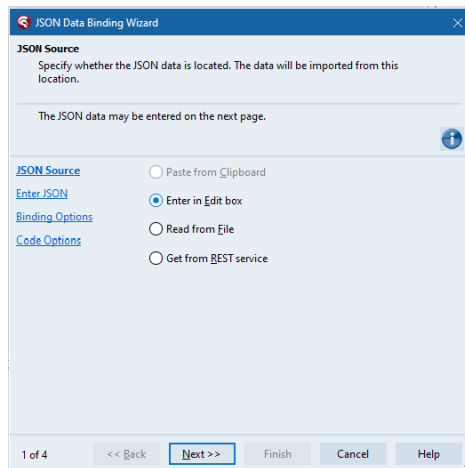


All About Data FireDAC & Web

JSON Data Binding Wizard

New Wizard to create Delphi classes mapped to JSON data structures

- Starts from JSON text
- Supported libraries are REST.Json and System.JSON.Serializers (more can be added)
- Multiple “Binding Options” specify how to map JSON elements to Delphi elements
- “Code Options” specify how to name Delphi entities, how to use properties



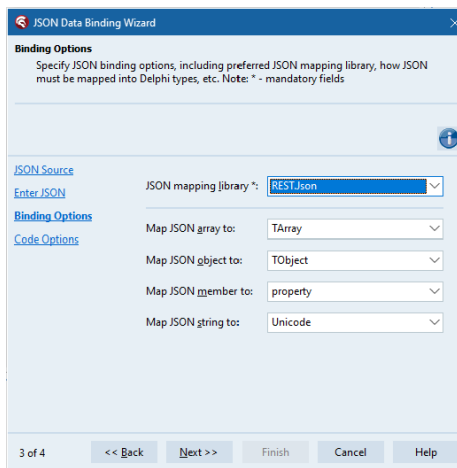
JSON Data Binding Wizard

JSON Source
Specify whether the JSON data is located. The data will be imported from this location.

The JSON data may be entered on the next page.

[JSON Source](#) ☐ Paste from Clipboard
[Enter JSON](#) ☒ Enter in Edit box
[Binding Options](#) ☐ Read from File
[Code Options](#) ☐ Get from REST service

1 of 4 << Back Next >> Finish Cancel Help



JSON Data Binding Wizard

Binding Options
Specify JSON binding options, including preferred JSON mapping library, how JSON must be mapped into Delphi types, etc. Note: * - mandatory fields

[JSON Source](#)
[Enter JSON](#)
[Binding Options](#)
[Code Options](#)

JSON mapping library*: REST.Json

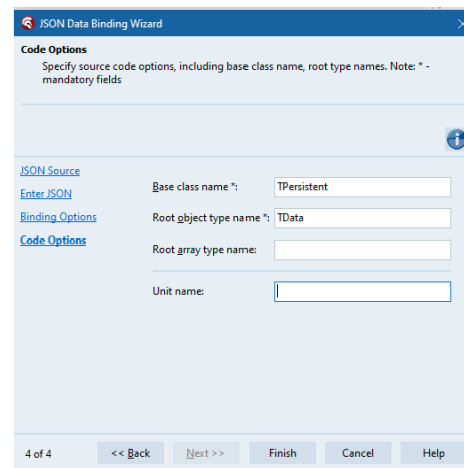
Map JSON array to: TArray

Map JSON object to: TObject

Map JSON member to: property

Map JSON string to: Unicode

3 of 4 << Back Next >> Finish Cancel Help



JSON Data Binding Wizard

Code Options
Specify source code options, including base class name, root type names. Note: * - mandatory fields

[JSON Source](#)
[Enter JSON](#)
[Binding Options](#)
[Code Options](#)

Base class name*: TPersistent

Root object type name*: TData

Root array type name:

Unit name:

4 of 4 << Back Next >> Finish Cancel Help

JSON Data Binding Wizard

```
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874",
    "geo": {
      "lat": "-37.3159",
      "lng": "81.1496"
    }
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
}
```

```
uses uJsonMapping, System.JSON;

procedure TForm1.Button1Click(Sender: TObject);
begin
  RESTRequest1.Execute;

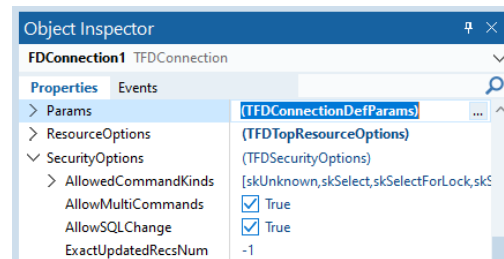
  TJSONMapper<Users>.SetDefaultLibrary('System.JSON.Serializers');
  var LUsers := Users.FromJSON(RESTResponse1.JSONText);

  Memo1.Clear;
  for var i: integer := Low(LUsers.Dataset) to High(LUsers.Dataset) do
  begin
    Memo1.Lines.Add(LUsers.Dataset[i].name);
    Memo1.Lines.Add(LUsers.Dataset[i].username);
    Memo1.Lines.Add(LUsers.Dataset[i].email);
    Memo1.Lines.Add(LUsers.Dataset[i].address.street);
    Memo1.Lines.Add(LUsers.Dataset[i].address.suite);
    Memo1.Lines.Add(LUsers.Dataset[i].address.city);
    Memo1.Lines.Add(LUsers.Dataset[i].address.zipcode);
    Memo1.Lines.Add(LUsers.Dataset[i].address.geo.lat.ToString);
    Memo1.Lines.Add(LUsers.Dataset[i].address.geo.lng.ToString);
    Memo1.Lines.Add(LUsers.Dataset[i].phone);
    Memo1.Lines.Add(LUsers.Dataset[i].website);
    Memo1.Lines.Add(LUsers.Dataset[i].company.name);
    Memo1.Lines.Add('-----');
  end;
end;
```


FireDAC More Secure Coding

FireDAC SecurityOptions property: new features to help write more secure code

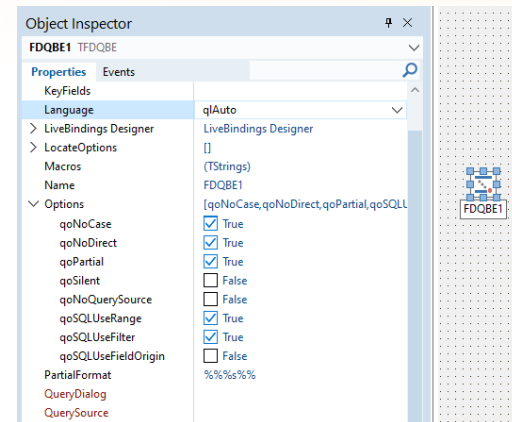
- AllowedCommandKinds: restricts the categories of SQL operations a query can perform
- AllowMultiCommands: disables the execution of multiple SQL commands in a single query (if the underlying DB supports it)
- AllowSQLChange: can prevent the code to change the text of a FireDAC SQL query at runtime
- ExactUpdatedRecsNum: if > 0, enables extra checked on the number of records updated by a query execution



These features can help, but writing secure database applications still requires developers to follow a number of best practices

Data and FireDAC

- Added also several JSON streaming improvements
- FireDAC support for QBE (query by example)
 - Use data aware UI controls so that an end user can define data filters
- IBLite/IBToGo for iOS Simulator
- DB RTL: improved TBlobField display logic
- SpellChecking in TDBRichEdit
- FireDAC SQLite Version Update (SQLite 3.42)
 - No encryption support outside of the paid SQLite EE
 - In alternative, use version 3.31.1 with FireDAC encryption (FDE)



RAD Server and Sqids

- Sqids are an improved version of Hashids, see <https://sqids.org>
- Replace physical IDs in a URL with a series of letters
 - Similar to YouTube video IDs
 - Uses a standard algorithm available in many programming languages
 - Can be customized via properties
 - Includes a list of illegal words (multi language, but you can provide yours)
- Display record with ID 12345:
 - From /customer/12345 to /customer/tyyksole
- Fully integrated and ready-to-use in RAD Server
 - When a TRESTRequest.Resource property parameter name starts with "#" the value will be Sqids encoded
- Available as a standalone feature (class TSqidsEncoding)

HTTP, REST, and More on RAD Server

HTTP and REST Client Library

- Added support for HTTP DELETE method with content
- Improved redirect and cookie management
- Support for large files in TMultiPartFormData
- RESTRequest accepts multi-part ContentType
- Curl support on Windows and macOS (via libcurl)

RAD Server

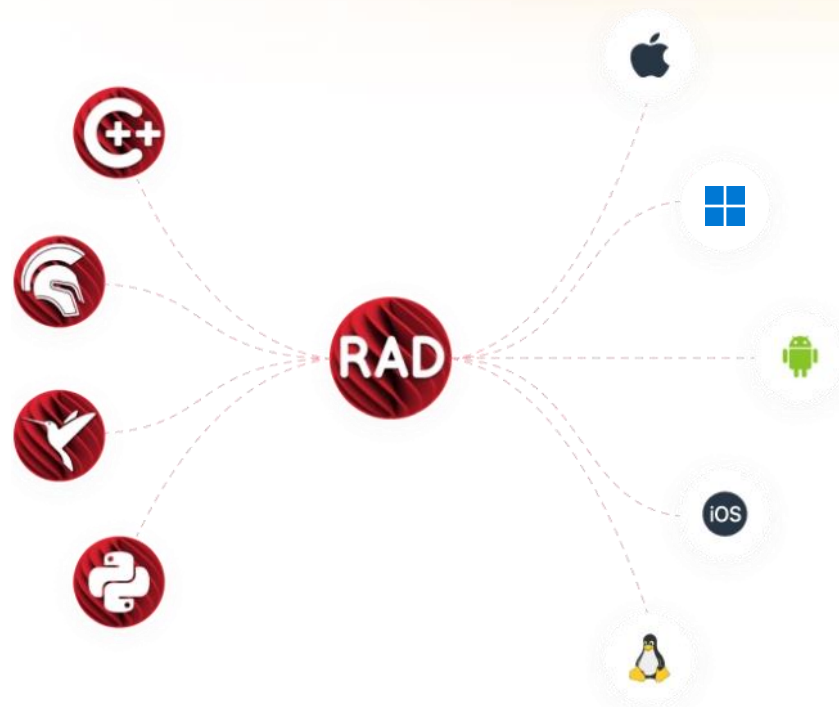
- Performance improvements by embedding FastMM5
- Paging allows page size override in client request
- Session authentication improvements



Delphi Language and RTL

Delphi Target Platforms

- Android 14
 - Android 10, 11, 12, 13
- iOS 17
 - iOS 16, no debugging support for iOS 17
- macOS Sonoma
 - macOS Ventura and Monterey
- Windows 11
 - Also Windows 10, 7 (SP1 +)
 - Windows Server 2019 and 2022
- Ubuntu 22
 - Ubuntu 20.04, RedHat 8, WSL2



More Delphi Language Improvements

NativeInt as a Weak Alias

- NativeInt becomes a “weak alias” of Integer or Int64, depending on the platform
- You cannot use it any more as a separate type: You get overload error on declaration, not ambiguous error on call
- Related: Further improvements in RTL support for Win64 types as needed

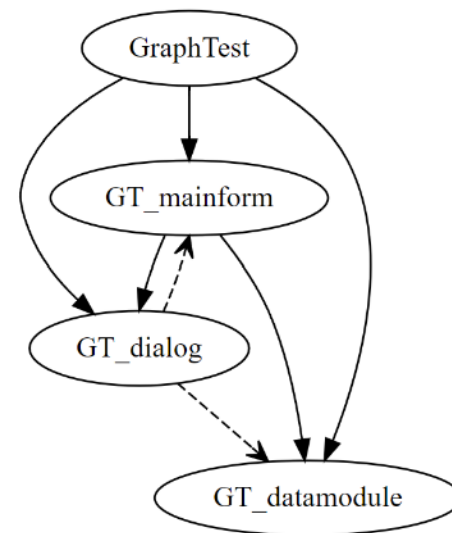
Support **NaN comparisons** as required by IEEE (Win32 compiler)

Improved warnings in generic classes (like in the non-generic counterpart)

Circular Uses Statements in Delphi

GraphViz representation

- Helps understanding “implementation” circular unit references, which put some stress on the compiler
- Options to export a GraphViz file with the complete unit dependency (including circular references in the implementation, marked as dashed lines)
- You can exclude unit families



Delphi Support for Android API 33

- Android API Level 33
 - Required for new apps published on Google Play Store
- Full support for latest Android
 - Including updates to some core Android APIs
 - Android Scoped Storage model support
 - IFMXCameraService and IFMXTakenImageService updated for this
 - IFMXExtendedClipboardService updated to prevent sensitive content
 - Cleaned up Uses Permissions page
 - New Android Splash Screen
 - Google Play Billing library 6.0.1
 - New built-in .dex.jar files

Complete Windows API Integration

Microsoft has (finally) provided complete metadata for the classic Windows API in WinMD format

- WinMD metadata available at <https://github.com/microsoft/win32metadata>
- We used it to generate over 300 units with 41 MB of Delphi code
- Available in GetIt

We have also refreshed to the latest version:

- The WinRT / Windows App SDK API
- The WebView 2 control API































Windows API from WinMD 1.0 by Embarcadero Technologies
Delphi headers of the complete Windows API, imported from the new WinMD public definitions from Microsoft. For 32 and 64 bit.

12 Sep 2023

Embarcadero Software License

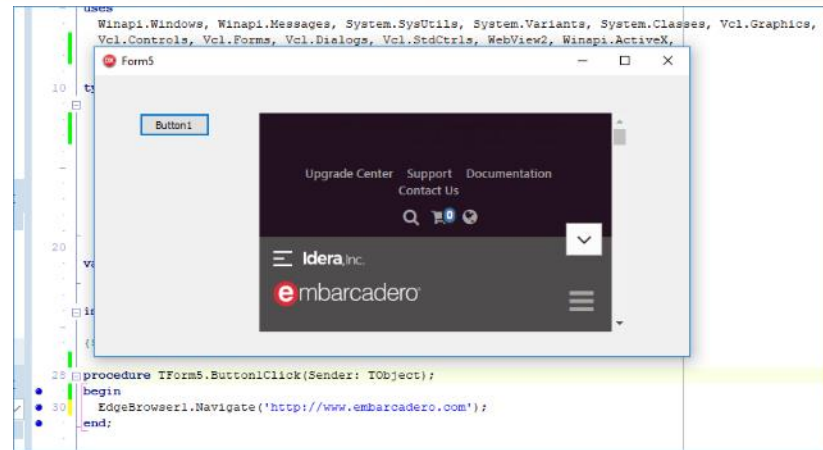
+ Install

Name	Date modified	Type	Size
 Windows.Graphics.Direct3D.pas	9/9/2023 7:51 PM	Delphi Source File	70 KB
 Windows.Graphics.Direct3D9.pas	9/9/2023 7:51 PM	Delphi Source File	137 KB
 Windows.Graphics.Direct3D9on12.pas	9/9/2023 7:51 PM	Delphi Source File	4 KB
 Windows.Graphics.Direct3D10.pas	9/9/2023 7:51 PM	Delphi Source File	293 KB
 Windows.Graphics.Direct3D11.pas	9/9/2023 7:51 PM	Delphi Source File	623 KB
 Windows.Graphics.Direct3D11on12.pas	9/9/2023 7:51 PM	Delphi Source File	6 KB
 Windows.Graphics.Direct3D12.pas	9/9/2023 7:51 PM	Delphi Source File	485 KB
 Windows.Graphics.DirectComposition.pas	9/9/2023 7:51 PM	Delphi Source File	107 KB
 Windows.Graphics.DirectDraw.pas	9/9/2023 7:51 PM	Delphi Source File	281 KB
 Windows.Graphics.DirectManipulation.pas	9/9/2023 7:51 PM	Delphi Source File	49 KB
 Windows.Graphics.DirectWrite.pas	9/9/2023 7:51 PM	Delphi Source File	246 KB
 Windows.Graphics.Dwm.pas	9/9/2023 7:51 PM	Delphi Source File	29 KB
 Windows.Graphics.DXCore.pas	9/9/2023 7:51 PM	Delphi Source File	14 KB
 Windows.Graphics.Dxgi.Common.pas	9/9/2023 7:51 PM	Delphi Source File	6 KB
 Windows.Graphics.Dxgi.pas	9/9/2023 7:51 PM	Delphi Source File	159 KB
 Windows.Graphics.Gdi.pas	9/9/2023 7:51 PM	Delphi Source File	411 KB
 Windows.Graphics.Hlsl.pas	9/9/2023 7:51 PM	Delphi Source File	1 KB
 Windows.Graphics.Imaging.D2D.pas	9/9/2023 7:51 PM	Delphi Source File	3 KB
 Windows.Graphics.Imaging.pas	9/9/2023 7:51 PM	Delphi Source File	174 KB
 Windows.Graphics.OpenGL.pas	9/9/2023 7:51 PM	Delphi Source File	196 KB
 Windows.Graphics.Printing.pas	9/9/2023 7:51 PM	Delphi Source File	417 KB
 Windows.Graphics.Printing.PrintTicket.pas	9/9/2023 7:51 PM	Delphi Source File	9 KB
 Windows.Management.MobileDeviceMa...	9/9/2023 7:51 PM	Delphi Source File	19 KB
 Windows.Media.Audio.Apo.pas	9/9/2023 7:51 PM	Delphi Source File	49 KB
 Windows.Media.Audio.DirectMusic.pas	9/9/2023 7:51 PM	Delphi Source File	53 KB
 Windows.Media.Audio.DirectSound.pas	9/9/2023 7:51 PM	Delphi Source File	57 KB
 Windows.Media.Audio.Endpoints.pas	9/9/2023 7:51 PM	Delphi Source File	20 KB
 Windows.Media.Audio.pas	9/9/2023 7:51 PM	Delphi Source File	295 KB

Windows Platform Integration

Edge Browser

- UserAgent is available in ICoreWebView2Settings
- ICoreWebView2Profile2 contains methods ClearBrowsingData, ClearBrowsingDataAll, and ClearBrowsingDataInTimeRange
- New OnDownloadStarting event
- New NavigateWithWebResourceRequest method
- Edge component surfaces the Print and ShowPrintUI operations



Changes to Floating-Point Exceptions

- Disabling Floating-Point Exceptions on All Platforms
 - The RTL used floating-point number exceptions since the early days
 - Modern operating systems and their official dev tools, including Windows, macOS, and Linux, decided not to use the FP exception model
 - This is causing a mismatch with platform libraries, we are solving by adapting our model
 - Old behavior can be restored via `System.math.SetExceptionMask`



Quality and Stability

Quality Improvements and Feature Extensions in All Subsystems

Extensive quality work and many small improvements, in

- IDE
- Debuggers
- Platform toolchains and compilers
- Delphi LSP
- Delphi RTL
- FireDAC database
- FireMonkey
- VCL
- RAD Server... and all other subsystems in RAD Studio

Quality, Quality

- RAD Studio 12.0 is providing a **fix** for:
 - **1,027** issues reported by customers on Quality Portal (quality.embarcadero.com)
 - 877 bug reports
 - 150 feature requests
 - 1,809 issues in total including internal and public reports and features
 - 1,471 bugs reports
 - 338 feature requests

This doesn't include duplicates, issues that no longer apply or we decided not to fix. The overall total of QP issues addressed is **1,264**

A mega-webinar on
RAD Studio, Delphi & C++Builder 12.0

Q & A

Fernando Rizzato

Lead Pre-sales Consultant LATAM

Embarcadero Technologies

fernando.rizzato@embarcadero.com