11 Alexandria - Release 2

Go Up to Main Page

Go Up to What Was New in Past Releases

Updated RAD Studio 11.2 Alexandria release available (September 7th, 2022).

> ***Note:***
>
> - *See what is included in version **11 Alexandria***
> - *See what is included in version **11 Alexandria - Release 1***
> - *See what is included in version **11 Alexandria - Release 1.5***

Key Enhancements by product area in 11.2

RAD Studio 11 Alexandria - Release 2 (also known as 11.2) is available for installation. RAD 11.2 builds on the feature set of 11 and 11.1, enhancing existing features throughout the product and adding new capabilities. Delphi 11, C++Builder 11, and RAD Studio 11 are available to download to any active Update Subscription customer.

RAD Studio 11.2 strongly focuses on quality improvements. Key quality focus areas include:

- iOS Simulator for Delphi
- Markdown support
- VCL, FMX, and RTL Quality
- RTL Performance
- Delphi Code Insight (DelphiLSP)
- C++ Code Insight (cquery)
- IDE High DPI, VCL form designer, and Quality
- GetIt
- Android Target API
- FireDAC, including FDMonitor and FDExplorer
- Delphi and C++ Toolchain Quality

Contents

- Debugger Quality
- RAD Server

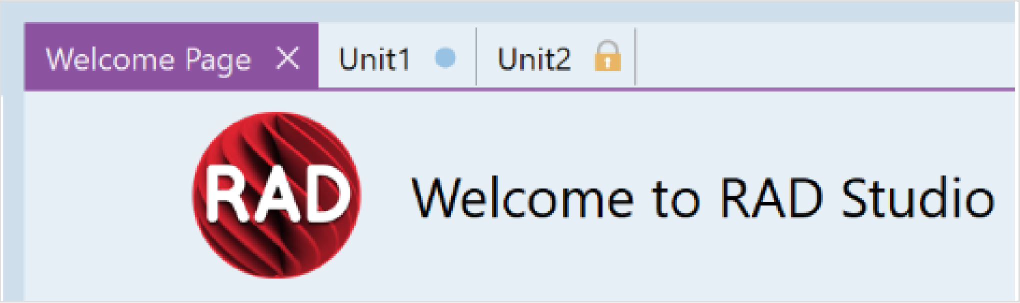
IDE Enhancements

High DPI VCL Designer

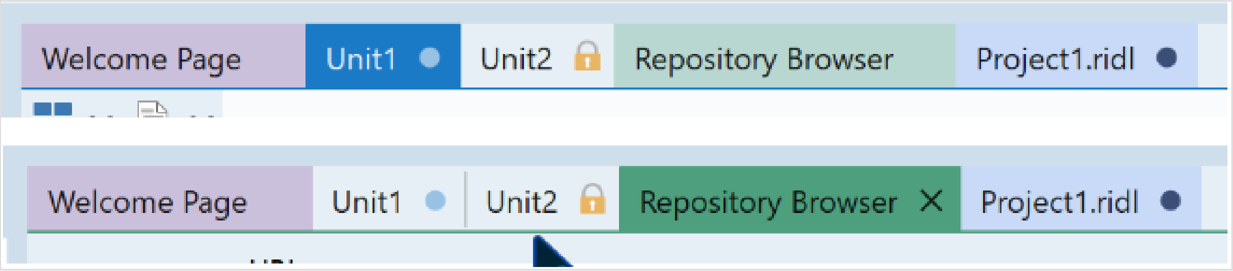
The VCL designer now uses a Windows-like style when designing, meaning controls in the designer always draw using this style unless this feature is disabled. This style matches the Windows light or dark theme, whichever Windows is currently using. RAD Studio added support for displaying VCL styles in the designer, intended for designing using the style your app would use at runtime, in 11.0. At the time, that option was off by default. Because the VCL designer now uses a style, using VCL styles in general in the designer is now on by default for new installations. If upgrading to 11.2, you may need to turn it on. You can turn off both using styles for the default Windows look and feel and using styles in the VCL designer at all, in the Options > User Interface > Form Designer page, '**Enable VCL Styles**' and '**Mimic the system style**'.

Editor Tabs

In RAD Studio version 11.2, some kinds of tabs, such as source control or COM-type library editors, are given different colors to be easily differentiated from normal code tabs. You can control if tabs may display customer colors via the new tab colors configuration.



There is support for custom tab colors via the ToolsAPI in the new INTACustomEditorView280 interface.



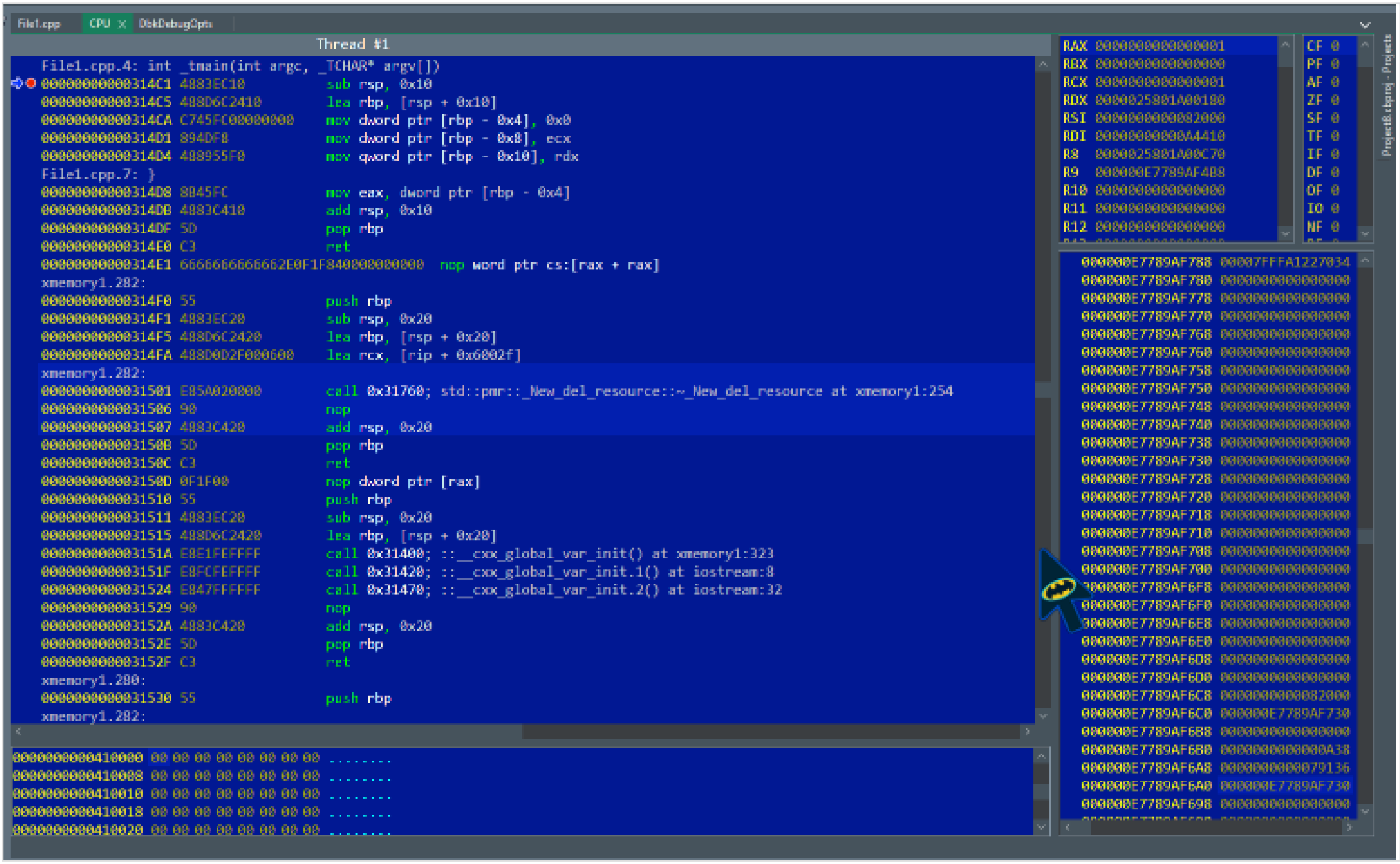
The 'X' close button on tabs is now optional. There is a new settings page, IDE Options > User Interface > Editor > Tabs, where users can edit the 'X' buttons to be visible: always, never, or for the active tab.

Tabs opened while debugging now also render differently, with the caption in italics.

Tabs for read-only files now display a padlock icon in the same space where the Modified dot would be displayed. In addition, the behavior of the Read-Only setting in the editor is now configurable. By default, it matches the file system status in order to prevent issues accidentally editing RTL/VCL/FMX/include, etc., files. This release brings the new option, 'Read-only status matches file system': When unchecked, the IDE reverts to the old behavior where the status was not correlated to the read-only flag for the file on disk, and you could mark it modifiable and edit in the editor even if the file on disk is read-only.

CPU View Syntax Highlighting

In RAD Studio 11.2, the CPU view now syntax-highlights the assembly code onscreen. Users can change this in IDE Options > Debugger > Disassembly and choose between no extra coloring (the old behavior), using the IDE theme colors or the editor colors.



```
File1.cpp.4: int _tmain(int argc, _TCHAR* argv[])
00000000000314C1 4883EC10           sub rsp, 0x10
00000000000314C5 488D6C2410         lea rbp, [rsp + 0x10]
00000000000314CA C745FC00000000     mov dword ptr [rbp - 0x4], 0x0
00000000000314D1 894DF8             mov dword ptr [rbp - 0x8], ecx
00000000000314D4 488955F0           mov qword ptr [rbp - 0x10], rdx
File1.cpp.7: }
00000000000314D8 8B45FC             mov eax, dword ptr [rbp - 0x4]
00000000000314DB 4883C410           add rsp, 0x10
00000000000314DF 5D                 pop rbp
00000000000314E0 C3                 ret
00000000000314E1 6666666666662E0F1F840000000000  nop word ptr cs:[rax + rax]
xmemory1.282:
00000000000314F0 55                 push rbp
00000000000314F1 4883EC20           sub rsp, 0x20
00000000000314F5 488D6C2420         lea rbp, [rsp + 0x20]
00000000000314FA 488D0D2F000600     lea rcx, [rip + 0x6002f]
xmemory1.282:
0000000000031501 E85A020000         call 0x31760; std::pmr::_New_del_resource::~_New_del_resource at xmemory1:254
0000000000031506 90                 nop
0000000000031507 4883C420           add rsp, 0x20
000000000003150B 5D                 pop rbp
000000000003150C C3                 ret
000000000003150D 0F1F00             nop dword ptr [rax]
0000000000031510 55                 push rbp
0000000000031511 4883EC20           sub rsp, 0x20
0000000000031515 488D6C2420         lea rbp, [rsp + 0x20]
000000000003151A E8E1FEFFFF         call 0x31400; ::__cxx_global_var_init() at xmemory1:323
000000000003151F E8FCFEFFFF         call 0x31420; ::__cxx_global_var_init.1() at iostream:8
0000000000031524 E847FFFFFF         call 0x31470; ::__cxx_global_var_init.2() at iostream:32
0000000000031529 90                 nop
000000000003152A 4883C420           add rsp, 0x20
000000000003152E 5D                 pop rbp
000000000003152F C3                 ret
xmemory1.280:
0000000000031530 55                 push rbp
xmemory1.282:
```

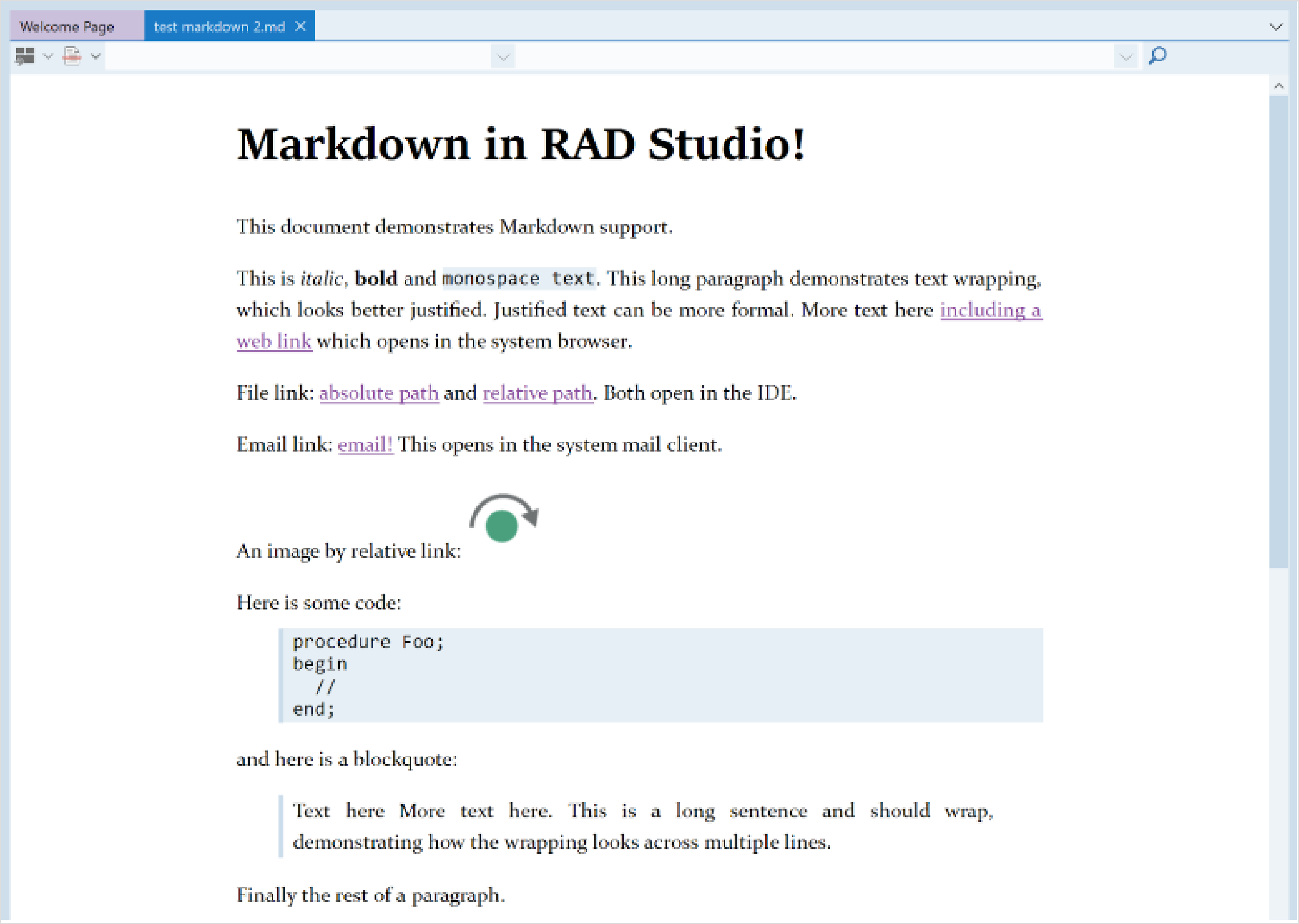
```
RAX 0000000000000001   CF 0
RBX 0000000000000000   PF 0
RCX 0000000000000001   AF 0
RDX 0000025801A00180   ZF 0
RSI 0000000000082000   SF 0
RDI 00000000000A4410   TF 0
R8  0000025801A00C70   IF 0
R9  000000E7789AF408   DF 0
R10 0000000000000000   OF 0
R11 0000000000000000   IO 0
R12 0000000000000000   NF 0
```

```
000000E7789AF788 00007FFFA1227034
000000E7789AF780 0000000000000000
000000E7789AF778 0000000000000000
000000E7789AF770 0000000000000000
000000E7789AF768 0000000000000000
000000E7789AF760 0000000000000000
000000E7789AF758 0000000000000000
000000E7789AF750 0000000000000000
000000E7789AF748 0000000000000000
000000E7789AF740 0000000000000000
000000E7789AF738 0000000000000000
000000E7789AF730 0000000000000000
000000E7789AF728 0000000000000000
000000E7789AF720 0000000000000000
000000E7789AF718 0000000000000000
000000E7789AF710 0000000000000000
000000E7789AF708 0000000000000000
000000E7789AF700 0000000000000000
000000E7789AF6F8 0000000000000000
000000E7789AF6F0 0000000000000000
000000E7789AF6E8 0000000000000000
000000E7789AF6E0 0000000000000000
000000E7789AF6D8 0000000000000000
000000E7789AF6D0 0000000000000000
000000E7789AF6C8 0000000000082000
000000E7789AF6C0 000000E7789AF730
000000E7789AF6B8 0000000000000000
000000E7789AF6B0 0000000000000A38
000000E7789AF6A8 0000000000079136
000000E7789AF6A0 000000E7789AF730
000000E7789AF698 0000000000000000
```

```
0000000000410000 00 00 00 00 00 00 00 00 ........
0000000000410008 00 00 00 00 00 00 00 00 ........
0000000000410010 00 00 00 00 00 00 00 00 ........
0000000000410018 00 00 00 00 00 00 00 00 ........
0000000000410020 00 00 00 00 00 00 00 00 ........
```

Markdown Support

The RAD Studio 11.2 IDE now supports Markdown (.md) files and both Markdown and HTML richly formatted previews. Opening a Markdown file will show a rich rendered preview of the file. Markdown and HTML can only be edited in the code editor, not the formatted preview.



Markdown in RAD Studio!

This document demonstrates Markdown support.

This is *italic*, **bold** and `monospace text`. This long paragraph demonstrates text wrapping, which looks better justified. Justified text can be more formal. More text here including a web link which opens in the system browser.

File link: absolute path and relative path. Both open in the IDE.

Email link: email! This opens in the system mail client.

An image by relative link:

Here is some code:

```
procedure Foo;
begin
  //
end;
```

and here is a blockquote:

> Text here More text here. This is a long sentence and should wrap, demonstrating how the wrapping looks across multiple lines.

Finally the rest of a paragraph.

Markdown files support:

- Email and Web links, which open in the system default email client or web browser
- File links, which open in the IDE
- Images
- Tables

Plus other standard Markdown formatting.

All image and file links can be either **absolute** or **relative** to the location of the Markdown file. Currently, only local paths are supported. If your Markdown file refers to a web link, such as loading an image from a server, it will not be loaded.

Markdown files can be set as the default readme for a project, meaning the Markdown file will be displayed when the project opens. You can configure this for a project via the Project > Project Page Options menu.

Removed Internet Explorer

In RAD Studio 11.2, the embedded Internet Explorer control was fully removed throughout the IDE. The rich HTML preview, the Markdown support, and HelpInsight all use a native VCL HTML control.

Font and Styles

Previous releases supported changing the IDE main window's font and font size via editing some registry keys. RAD Studio 11.2 offers the option to change the IDE UI font and size in the IDE Options > User Interface page in the '**IDE Font**' section. Similarly, the UI has now re-enabled a previously hidden setting to let the IDE use a custom style. The Options > User Interface > IDE Style page lets you use a custom VCL style in addition to the Light, Dark, and Mountain Mist styles. Both these are primarily intended to assist accessibility.

Projects View

The 11.2 release changed the behavior of double-clicking an item in the Projects view so that it only expands/collapses a tree entry if it represents a folder. Now, double-clicking performs the following actions:

- Double-clicking on a unit opens the unit.
- Double-clicking on a folder expands or collapses it.
- Double-clicking on a project activates the project if it is not the currently active project. If it is a currently active project, it will expand/collapse it.
- Double-clicking a C++ form selects and opens the form without expanding the node.

Effectively, double-clicking should do the 'right' or expected action by default instead of always simply expanding the tree node.

GetIt Enhancements

In RAD Studio 11.2, the GetIt package manager dialog allows users to filter by Language. This dialog is only displayed if using RAD Studio (not Delphi or C++Builder) with both personalities installed. The new filter allows you to list only Delphi packages or only C++Builder packages. See more at Tools> GetIt Package Manager.

The package manager dialog now allows you to install a package from a local file and online packages. To do so, use the '**Load Local Package**' button under the filters on the bottom left of the GetIt dialog. This lets you download and install a package on a computer not connected to the internet, develop a GetIt package before publishing it, and simplify the steps for installing packages among multiple developers in a team. This feature is only supported with an active update subscription.

Welcome Page

- The Welcome page now contains a restored 'New In GetIt' pane. Displaying the most recently modified packages in GetIt and facilitating the discovery of new or recently updated libraries and packages.
- The Welcome Page will, by default, reopen when all projects are closed.

Other IDE Improvements

- The Reopen list is now populated when a project is opened, not closed, meaning if the IDE is terminated before closing a project, you will see your most recently opened project visible.
- There are several performance improvements and general quality fixes when running the IDE under Remote Desktop.
- The Compile progress dialog and the Updating Local File Cache dialog (used when updating a remote SDK) now format the lines of code and number of bytes using the local locale, i.e., with commas or other separators. Making the numbers easier to read.
- To find items easier, the options in **File > New > Other > Inheritable Items** are now sorted.
- We use 'Options' in the IDE for changing settings (e.g., the IDE and Project 'Options' dialogs), but several menu items that opened options used 'Properties' instead. These have been renamed, so now there are: 'Tab Options' on the editor tabs context menu, 'Editor Options' on the editor context menu, and a new 'Status Bar Options' for the editor status bar.
- There are also various quality fixes across the IDE, especially for high DPI.

LSP Improvements (Delphi and C++)

Delphi LSP

Marking inactive code

Starting RAD Studio 11.2, the code editor will now render code between inactive IFDEF-s, i.e., code that is conditionally compiled out / not compiled, differently from code that will be compiled.

Inactive code is drawn blended with, or semitransparent to, the background color. All code and editor markup in an inactive region, whether code or features like brace highlights or selections, is subdued compared to active code regions. Inactive code can be as short as one character but often spans multiple lines.

Because DelphiLSP provides this feature, it will reflect the LSP server's knowledge of defined macros `($define-s.)` This feature also works when using DelphiLSP with VSCode via the DelphiLSP addon.

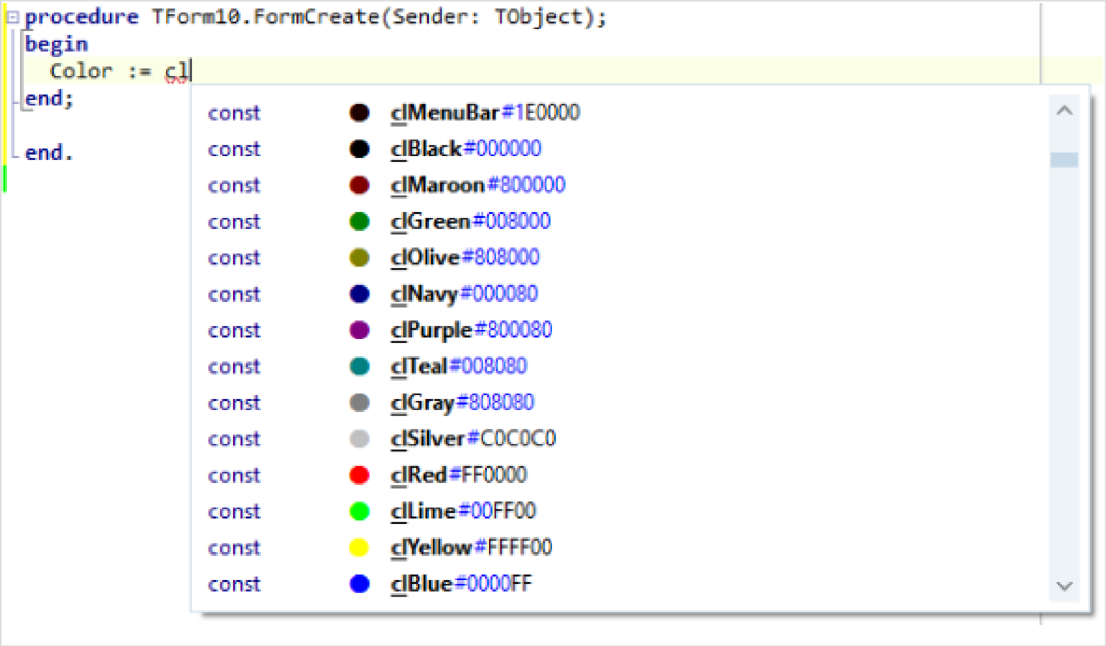
[[File:InactiveCode.png|link=|border]

XSLT Transforms for Help Insight

In RAD Studio 11.2 includes improvements to the Help Insight. Using Delphi LSP, HelpInsight tooltips are generated by running an XSLT transform over the XML. You can edit HelpInsight.xsl to modify this transform. Note that HTML inside the XML should be transferred through: this work aims to restore the feature as it was pre-10.4.

Color constants shown in Code Completion and Tooltip Insight

When completing a VCL or FMX color constant (such as clWindow, clRed, etc.), the code completion window will show a preview of the color and its hex value.



```
procedure TForm10.FormCreate(Sender: TObject);
begin
  Color := cl
end;

end.
```

```
const   ●  clMenuBar#1E0000
const   ●  clBlack#000000
const   ●  clMaroon#800000
const   ●  clGreen#008000
const   ●  clOlive#808000
const   ●  clNavy#000080
const   ●  clPurple#800080
const   ●  clTeal#008080
const   ●  clGray#808080
const   ●  clSilver#C0C0C0
const   ●  clRed#FF0000
const   ●  clLime#00FF00
const   ●  clYellow#FFFF00
const   ●  clBlue#0000FF
```

The same color is also shown in the tooltip when hovering over a color constantly.

C++ and cquery

Significant Cquery performance improvements, such as greatly improved performance, improving projects and project groups management, and limiting CPU consumption, are included in 11.2 based on the C++-only 11.1.5 release.

C++ LSP

In the IDE Tools Options, under Tools > Options > User Interface > Editor > Language there is a tab called "LSP Behavior" on the C++ LSP page. This has been available since 11.0, but we have a new third setting, 'Index all files in the project group.' For more information, please read the Configuring C++ Code Insight documentation to tune its behavior for your projects and needs.

Compilers and Toolchain Improvements

Android Target API

This release targets Android API Level 32 (up from API level 30 in 11.1). The main reason is Google Play store mandates Android API Level 31 starting in August 2022 for new submissions and in November 2022 for updates.

There are several related changes to the Android toolchain for both Delphi (32-bit and 64-bit) and C++Builder (32-bit), including:

- Updating the JDK version installed with RAD Studio to JDK 11 (required by the latest version of the Android SDK and tools).
- Upgrading the default value for the 'targetSdkVersion' manifest attribute to 32.
- Updating the 'AndroidManifest.xml' template to set a value for the 'exported' attribute in the declaration of the 'FMXNativeActivity' activity (required by API Level 31 support).

Rad Studio version 11.2 now supports generating a PKCS12 Keystore and also includes new permissions in the Application > Uses Permissions page of the Project Options dialog. And an update to the Android Java SDK finder so that if the Eclipse Adoptium JDK is available, it will be used in preference to other JDKs.

Updated AndroidManifest.template.xml file

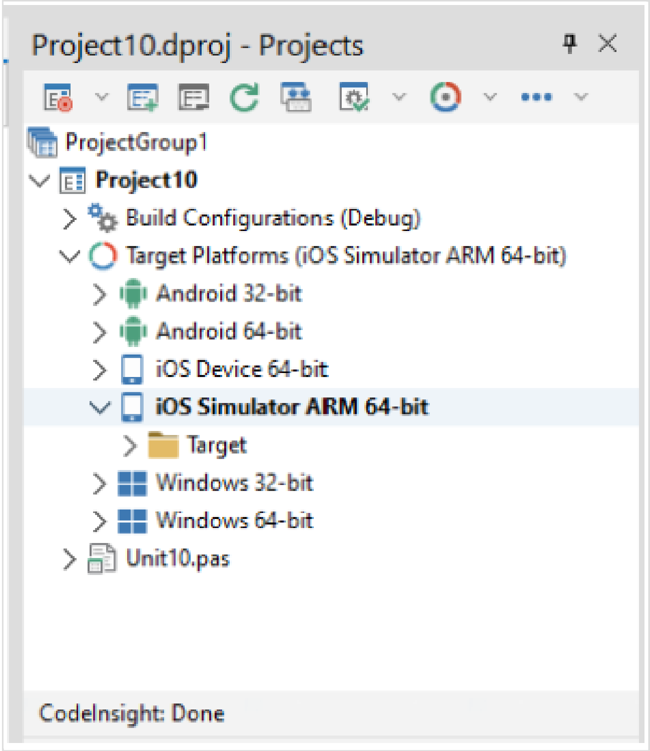
If you have an existing FireMonkey application project for the Android platform from a previous RAD Studio version, you need to delete the AndroidManifest.template.xml file from your project´s root directory. Please follow the steps on the AndroidManifest page to ensure that the AndroidManifest.template.xml file is setting a value for the exported attribute in the declaration of the FMXNativeActivity activity.

Support for Delphi iOS Simulator

RAD Studio 11.2 offers iOS Simulator support for Delphi. The support includes only the ability to generate iOS Simulator binaries for macOS devices running on ARM-64 (M1 or M2).

> ***Note:*** *Apple is phasing out Intel macOS, so RAD Studio does not support the iOS Simulator on Intel macOS hardware and instead focuses on the ARM-64 target.*

The RAD Studio IDE can compile, deploy, execute and debug applications targeting the iOS Simulator for ARM 64-bit.



Delphi Compiler Enhancements

As part of the RAD Studio 11.2 Delphi compiler improvements, Delphi further expands Address Space Layout Randomization (ASLR) support for the Win64 platform by supporting High Entropy ASLR (HE ASLR) by adding the ability to randomize memory addresses at high locations.

C++ Compiler and Toolchain improvements

This release includes quality improvements in the C++ toolchain. Some of the quality improvements include:

- Ilink64 has a number of fixes.
- The compiler will now give an error when exceeding the VIRDEF maximum. Previously, this would cause the linker to give an error.
- STL iostreams performance has been greatly improved and now takes less than 25% of the time.
- Improvements to the object converter (omf2coff.)
- Improvements to delay loading.
- A number of record helpers from Delphi are integrated into C++.
- A number of general quality improvements in the compiler and RTL.

Debuggers

LLDB for Win64 C++

You can now do remote debugging using the Win64 C++ debugger.

LLDB for Delphi

The 11.2 release includes support for debugging iOS64 and the iOS Simulator.

The LLDB Delphi support has many improvements to its understanding of the Delphi language, including basic support for sets, dynamic arrays, nil, pointer casts, empty strings, and more.

In 11.2, the LLDB debugger is used for the following platforms:

- C++ Win64
- Delphi macOS ARM 64
- Delphi macOS Intel 64
- Delphi Android 64
- Delphi and C++ iOS 64
- Delphi iOS ARM Simulator (ARM 64)
- Delphi Linux 64

This is part of our long-term strategy to move to LLDB.

LLDB for Delphi for Linux

This release supports the new LLDB for Delphi for the Linux 64bit platform. Linux64 previously used GDB, and there were many reported issues around evaluating anything other than basic types. We believe LLDB should be a great improvement.

To debug a Linux console application using standard input (e.g., ReadLn), see the documentation on using a launcher application.

Connecting Python 3

Running LLDB on Linux via PAServer requires Python 3 to be installed, and that LLDB can find it. If you get errors referring to Python, please see the Known Issues for info on configuring Python 3.

VCL and FireMonkey Changes

WebView2 in TEdgeBrowser and TWebBrowser

- The new TWebBrowser.EdgeUserDataFolder property lets you specify where user data (cookies, caches resources, etc.) are stored.
- The new TWebBrowser.EdgeBrowserExecutableFolder property lets your app find and use a specific WebView2 runtime instead of using the system's version.
- Updated the definition of the Web View 2 API (not binary compatible with the VCL components), including support for the ICoreWebView2_2 interface.

General VCL Improvements

This release focuses on quality and includes:

- Improved the TNumberBox input management
- Improved the TDateTimePicker format settings
- Using CardPanels inside a frame
- Deleting a TCardPanel
- Improved the PopupComponent property of a TButton with bsSplitButton
- Improved the TListView overlays when the OwnerData is True
- Improved High DPI issues for TBalloonHint.

FireMonkey

This release focuses on quality and performance over new features. Here are some of the key improvements:

- TBufferedLayout reacts to an update of the styles of contained controls
- Added hints to the buttons of the FMX style designer.
- Improved bitmap processing on iOS with Metal.
- Support for an iOS status bar with dark icons.
- 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
- TListBox now supports Shift+Click to select a range.

RTL and Data

FireDAC reFind Utility

The reFind utility can assist the migration of dbGo projects to FireDAC. In addition to a migration script, there is also a compatibility unit, FireDAC.ADO.Migrate.pas.

When performing code replacement with reFind.exe, you can use a new option to "leave the old target code as a comment instead of deleting it". Use the new /C switch to comment instead of removing code.

Updated FireDAC Drivers

FireDAC supports the latest Microsoft ODBC Driver for SQL Server (version 18).

Improved the MSSQL Driver from MacOS on M1 with better handling of `SQL_C_WCHAR`.

Added support for MongoDB's new transaction feature. The main MongoDB API is located in FireDAC.Phys.MongoDBWrapper.pas unit. The new transaction support is provided by the new API in:

- TMongoTransactionOptions class
- TMongoSession class
- TMongoConnection.NewSession method
- TMongoCollection.Session method

FDMonitor and FDExplorer

The FDMonitor tool was improved. In particular:

- FDExplorer / FDMonitor now has syntax highlighting in SQL views.
- FDMonitor has a new "**SQL Output**" view for tracking only SQL commands.
- FDMonitor has new options for tracking the "**SQL Output**" view.
- FDExplorer / FDMonitor now has Search / Search Next commands.
- Added new items to TFDMoniEventKind: ekSQL, ekSQLVarIn, ekSQLVarOut to produce output to "**SQL Output**".
- Optimization of ODBC drivers with tracing is enabled when ekVendor is excluded.
- Optimization of TFDConnectionDef loading.
- Font size on HighDPI was improved for both FDExplorer and FDMonitor.

In addition, the embedded FireDAC SQL text property editor now supports SQL syntax highlighting, line numbers, and support for common shortcut keys like Ctrl+F, Ctrl+R, F3, and Alt-G.

RTL Enhancements

The following are Delphi RTL improvements:

ZLib

Updated the integrated ZLib to 1.2.12 to address a recently patched vulnerability. Some other ZIP file behavior, including using “stored” and not deflated files, comment management, and the behavior of ZDecompressStream with truncated Zlib data, have been improved., System.Zip also can now preserve symlinks on Linux.

The performance of Zlib when writing small Data packages has been improved.

General

- There is a new TProxySubrangeStream class in System.Classes.pas. The new stream class represents a "subrange of another stream" and acts as a proxy.
- The PCRE regex engine compiles with JIT enabled, offering significantly better performance. Additionally, the library now exposes the procedure TPerlRegEx.Study in the TRegEx record.
- Several RTTI-related improvements include TVirtualMethodInterceptor with const parameters, invocation with "pascal" calling convention, and TVirtualInterface issue with float values on iOSDevice64.
- Optimized several core functions, including (among others) TStringHash, InternalUniqueStringA, and InternalUniqueStringU, plus two System unit functions, _UStrEqual and _UStrCmp, and SysUtils.CompareStr.
- System.SysUtils.StrToFloatDef no longer changes the numerical exception mask.

HTTP and REST Client Library

TRestRequest’s handling of MIME types and multipart/form-data has been improved. In addition, proxy detection has been improved for Windows, and THTTPNetClient SSL-certificate checks have been improved for Android.

RAD Server

Rad Server improvements:

- New auth token/session expiration for RAD Server. There is a new parameter in the RAD Server configuration (EMS.INI) called SessionTimeoutAdded new configuration options in emsserver.ini to specify the paths to FDConnectionDefs.ini and FDDrivers.ini FireDAC configuration files
- Made it possible to use TEMSFireDACClient.PostUpdates with a custom session token
- Improved the list function of TEMSFileResource
- Removed the incorrect 2-users limitation in RAD Server Lite
- Added the IP Address and DateTime to RAD Server logging
- Added an Execute context menu item to the TEMSFireDACClient component (similar to the TRESTRequest component) so that you can create persistent fields based on the result set.

Querying SKU and Platforms on the Command Line and via ToolsAPI

In RAD Studio 11.2, third-party components or plugin vendors can query what SKU is installed (e.g., Architect, Community, etc.), what platforms are installed (Win64, macOS ARM, etc.), what personalities are installed (Delphi or C++Builder or both), and if it is a Trial. Querying SKUs can be useful for making decisions when installing components, including in your installer, and changing behavior at designtime inside the IDE itself.

Run bds.exe with the new `/ProductInfo:KIND` parameter to get the console´s output. KIND can be Trial, SKU, Platforms, or Personalities. Learn more about the new parameter on the General options for BDS.EXE page.

The new IOTAVersionSKUInfoService interface allows you to query the list of installed platforms, the SKU, and if the IDE is running as a Trial version.

See Also

- Installation Notes
- Release Notes
- New features and fixed issues
- What Was New in Past Releases

Retrieved from "http://docwiki.embarcadero.com/RADStudio/Alexandria/e/index.php?title=11_Alexandria_-_Release_2&oldid=273784"