11 Alexandria - Release 3

Updated RAD Studio 11.3 Alexandria release available (February, 27th, 2023).

> ### *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*
> - *See what is included in version 11 Alexandria - Release 2*

Key Enhancements by product area in 11.3

RAD Studio 11 Alexandria - Release 3 (also known as 11.3) is available for installation. RAD Studio 11.3 allows users to customize the code editor look and feel with the new ToolsAPI and use the highlighting matching words function in the code editor. In addition, 11.3 introduces a new Biometric Authentication component for mobile platforms, plus many quality and performance enhancements in Delphi LSP and the different RAD Studio libraries.

RAD Studio 11.3 strongly focuses on quality improvements. Key quality focus areas include:

- Delphi Code Insight (DelphiLSP)
- Subversion DLLs
- IDE High DPI and the VCL form designer
- Delphi Compiler
- C++ Compiler and Toolchain
- Debugger Quality
- RTL Quality and Performance
- VCL
- FMX
- Data

Plus a few new features:

- ToolsAPI support for the Code Editor

Contents

IDE Enhancements

ToolsAPI support for the Code Editor

RAD Studio 11.3 offers users a complete, comprehensive, and very detailed set of Tools API interfaces that users can use to customize the Code Editor, such as painting, querying information about editors and their contents, getting events for the Code Editor, and more.

Some of the most exciting and useful plugins work within the editor. Therefore RAD Studio aims to provide a complete and valuable set of APIs, providing extensive coverage of code editor data and presentation. Not only does this provide an official supported API for plugins that previously required hacks to function, but the new API also makes it very easy to write plugins directly integrated into the Code Editor. It is such a powerful API that we have moved several editor functions over to use it. We hope the new API spurs innovation in plugins and productivity add-ons, and we're very keen to see what you and others come up with!

These APIs allow users to intercept all stages of painting in the editor, either by editor line, editor gutter, or editor text, and with events called both before and after each stage, and either add to what the IDE does or replace it.

For an editor, users can query its state, with a multitude of information available, and query a more detailed state for each line. Paint events include full context. Use this to do any conceivable editor painting or get any information you need about editors and what's present in them, even without painting.

The new ToolsAPI unit includes its classes, methods, and properties documentation. Explore and start using the new APIs:

- ToolsAPI
- ToolsAPI.Editor
- ToolsAPI.WelcomePage

We strongly encourage all third-party plugins or addons that draw in the code editor to use these new interfaces.

Highlighting Matching Words in the Code Editor

The RAD Studio Code editor has a new feature that allows you to see instances of the same word you have selected, or the cursor is present is highlighted on the screen. Words are matched anywhere onscreen, including in comments. In addition, the feature has the following options:

- Highlight matching words when a word is selected (double-click to select a word).
- Highlight matching words when the cursor is inside or adjacent to a word. Users can turn off the feature if required.
- Disable the highlight keywords as a match function with the UI setting. This is enabled by default.

```
{$IF DEFINED(CLR)}
          ParentForm := GetParentForm(LControl, False);
          if (FActiveControl <> LControl) and
            ((ParentForm = Self) or (ParentForm <> Screen.FActiveCustomForm)) then
            ParentForm.SetFocusedControl(LControl);
{$ELSE}

          ParentForm := GetParentForm(LControl);
          if ((FActiveControl <> LControl) or (LControl <> Screen.FActiveControl)) and
            (ParentForm <> nil) and
            ((ParentForm = Self) or (ParentForm <> Screen.FActiveCustomForm)) then
          begin
            if (ParentForm.ActiveControl <> nil) and (ParentForm.ActiveControl <> LControl) then
                ParentForm.ActiveControl.PerForm(WM_KILLFOCUS, 0, 0);
            ParentForm.SetFocusedControl(LControl);
          end;
{$ENDIF}
```

Delphi LSP

In the 11.3 release, RAD Studio aims to focus strongly on quality and resolve key issues.

Delphi LSP enhancements include a significant focus on quality. Key issues have been resolved. In addition, the following items have been added:

- The HelpInsight window uses a different method to calculate its size and matches the content much better now.
- Hovering over a static method implementation works as expected.
- `Ctrl+Left Click` navigation works as expected when using a read-only file.
- When pressing Enter inside an incomplete anonymous method, the IDE no longer adds an incorrect semicolon ("end;").
- An issue where Ctrl-Click-ing to navigate did not work when the area to navigate to was surrounded by compiler directives has been resolved. The parser can now handle simple conditional compiler directives such as:

```
{$IF CompilerVersion > 24 and Defined(Windows)}
```

> *Note: This does not affect parsing directives in the uses clause when units are added by the IDE, which uses a non-DelphiLSP system.*

- Code completing generic types inside a generic method declaration, both the result type and inside the generic `<>` type declaration, now works:

```
function Test<TResult>(a: T): TResult; // TResult now suggested in both places.
```

- Resolved an issue where DelphiLSP could not determine the path to the source code of a module if there were two or more different files with the same names in different projects.
- Resolved the issue where code completion did not work in an "if statement" that used `<` or `>` symbols in its condition.
- When completing a type declaration, private types declared within that type will now be shown in code completion.

Subversion

Starting this release, RAD Studio no longer ships the Subversion DLLs, requiring you to install Subversion if desired. Like Git and Mercurial, the IDE looks for their previously configured location. This means that the Subversion version is not correlated to the IDE version and can be kept up to date separately.

IDE High DPI, VCL form designer, and Quality

This release offers some improvements to the scaling of non-visual components, including in frames. The following are the main improvements:

- Resolved an issue where the position of non-visual components was incorrect when migrating from a pre-version 11 version of RAD Studio to an 11.x version.
- When the form scaling changes and the system scaling changes, non-visual components will retain the correct position and scaling.
- Resolved an issue with font sizes affecting frame scaling in high DPI mode, especially running the designer at low DPI.

Other IDE enhancements

- Users can now create a new Markdown file through **File > New**
- When the IDE is run with elevated privileges, the IDE now indicates this with '[Administrator]' at the start of the title bar caption.
- Locate the current build configuration's output file in Explorer by right-clicking the build config and choosing the **Show In Explorer** option.
- Replaced the editor structural flow icons (icons in line with the code indicating when a method returns, control flow breaks or moves, etc.) with high DPI versions.
- The opacity of disabled editor text (ifdef-ed out) can now be changed via a UI option on the **Editor > Search** page.
- The font size for the VCL designer's hint window has been improved.
- The icons for SyncEdit and structure flow in the editor have been replaced with high-DPI icons.
- The New Items dialog had an issue with Copy / Inherit / Use when the view was changed. This is now resolved.
- An issue where path dialogs did not show invalid paths in gray is resolved.
- Some stability issues using secondary editor windows have been resolved.
- Markdown files can now be opened in multiple editor windows simultaneously, and errors will not be displayed in the Structure view for Markdown files.

Compilers and Toolchains Improvements (Delphi and C++)

RAD Studio 11.3 comes with several quality improvements to most of the compilers shipped in RAD Studio.

Delphi Compiler

The 11.3 release fixed several compiler issues in different areas, including:

- Improvements for the Delphi LSP processing and performance enhancements.
- Fixes for the managed records and for issues with generics and constraints.
- Codegen improvements and corrections.
- Addressed a regression, which caused the compiler to report completely wrong Total Lines in the Compile Progress dialog. This was visible when building Spring4D.

C++ Compiler and toolchain

RAD Studio 11.3 brings some quality fixes like:

- Using a library suffix of `$(Auto)` now works for C++ projects.
- Resolved the issue where the dependency check for static libraries used the wrong name, causing it to be rebuilt unnecessarily.

Debuggers

The LLDB Debugger for Linux has been rebuilt with Ubuntu 18.x and Python 3.7.x, as its minimum requirements. Also, this release offers some fixed issues for macOS and Linux.

- In 11.3, all non-Windows platforms moved to LLDB. As a result, there is no support for GDB anymore.
- Formatting specifiers are supported in expression evaluation for LLDB.
- Resolved issues attaching to processes and loading symbols with LLDB.
- Resolved issues for Android32 around set evaluation and expression evaluation related to the switch to LLDB.
- New PAServer versions.
- Users can detach from a program using the C++ Win64 debugger.
- The inspector Range menu option is now available for arrays.
- Android64 supports naming threads.

Libraries Improvements

RAD Studio 11.3 has some improvements and many quality fixes in all of the libraries, Delphi RTL, C++ RTL, VCL, FireMonkey, FireDAC, the DB RTL, HTTP client libraries, and HTTP servers. In this section, we highlight the more significant changes.

Delphi RTL

Release 11.3 brings the following enhancements to Delphi RTL.

- Given that Android's versions 12 and 13 have modified the underlying implementation of the Android native memory manager's realloc C-API, RAD modified the pre-allocation logic (256 bytes are added for each length change for String and Dynamic-array if the memory footprint is larger than 64K bytes) to pre-allocate more than needed to reduce the realloc operating system calls and their copy operations.

> **Note:** If your code requires a lot of memory, you should revisit your app logic.

- The system RTL function `Move` has been optimized for better performance on modern CPUs, running 2 times to 5 times faster, depending on the actual CPU family and bitness.
- Updated ZLib library to version 1.2.13.
- Improved performance of _InitializeRecord.
- Improved performance of TDirectory.GetFiles for many files.

VCL

Release 11.3 brings the following enhancements to VCL.

- The TControlList VCL component now offers the ability to select multiple items using the following new properties and events:
  - property TCustomControlList.MultiSelect
  - property TCustomControlList.Selected
  - method TCustomControlList.ClearSelection
  - method TCustomControlList.SelectAll
- Addressed scaling issues in the TDBGrid, TComboBox, and TStringGrid controls, among others, and DPI change issues in several components, including TListView, TComboBox, and TStringGrid controls. Also, it improved the overall behavior when a form Scaled property is False.

- Fixed some TBalloonHint regressions.
- Improved StyleHook code to avoid possible Access Violations.
- TreeView graphical improvements on scaling and graphical objects rendering improvements for several other controls.
- Addressed a conflict in the TRichEdit control between SpellChecking and custom Popup menus.
- Improved NumberBox input with an active MinValue.
- Better support for TControlList controls with a very large number of (virtual) items.
- Resolved issues with the custom title bar button placement and size, including canvas size and the left caption margin.

FireMonkey

RAD Studio 11.3 offers the new TBiometricAuth component that provides the means of authorization via biometrics, such as Face ID, Fingerprints, on Android and iOS devices.

> **Note:** *For Android, the TBiometricAuth component is currently supported only on Android 10 or higher.*

Use TBiometricAuth from the component pallet to set the desired properties. If the application will be available for Android, ensure the prompt properties of at least PromptDescription and PromptTitle have values. Also, ensure to specify the desired strengths for the BiometricStrengths property. You can also create handlers for the OnAuthenticateSuccess or OnAuthenticateFail events to handle what should happen in those scenarios.

Explore and start using the new APIs:

- FMX.BiometricAuth

Release 11.3 also brings the following enhancements to FireMonkey.

- Added a new interface IFMXPhoneDialerListenerService service that intends to track changes in the state of the phone system service (Calls, Carrier, CallState).

> **Note:** *For Android only. Before directly tracking phone calls, developers must request permission* `android.permission.READ_PHONE_NUMBERS`*.*
>
> *If the end-user grants access, request service and invoke the* `Start` *method. Otherwise, the developer should explain why this permission is required.*
>
> *At the end of call tracking, call the* `Stop` *method.*

- FireMonkey apps now properly handle terminate requests from Windows Restart Manager.
- Added macOS ARM 64 to Radiant Shapes supported platform.
- Improved Firemonkey accessibility support for Windows when inspecting with Accessibility Insights and addressed an AV in FMX.ScreenReader.Win.pas.
- Several TWebBrowser improvements for the Windows platform, including the availability of 3D FMX apps and the implementation of TCustomWebBrowser.CaptureBitmap for Edge.
- Some iOS platform translation improvements for the iOS VirtualKeyboard Done button and iOS pickers buttons; iOS scrolling fixed for TDateEdit, TTimeEdit, TComboBox, and TiOSAddressBook.SaveContact.
- DocumentFile Provider support for Android.
- Android improvements when moving the cursor in a TMemo with finger slide, TAddressBook component.
- Windows improvements with TForm Cursor, TWinMultiDisplay.FindDisplay.
- FireMonkey graphics improvements in TBitmap.AssignFromSurface, TCanvas.TransformRect, TBitmap.Canvas.BeginScene, plus removed artifact lines in TMemo and TGridPanelLayout.

- Improvements and fixes for many controls, including Multiview (with controltype = platform), SpeedButton (isPressed when used in a TFrame), Label (FocusControl), TTreeViewItem's IsChecked, TEdit works inside a TVertScrollBar.
- ClipChildren now takes XRadius and YRadius into consideration.
- Improvements to Tab key cycling through all controls.

Data

Release 11.3 brings the following enhancements to the Data Library.

- Added preliminary support for UTF8 in database RTL (TStringField), FireDAC ODBC, DatS, or DataSet to better support ODBC drivers on Linux.
- FireDAC offers improvements for PostgreSQL performance. There is a new PostgreSQL connection setting FastFetchMode=Choose|Default|All|First, where:
  - **Choose** - lets FireDAC choose what the appropriate optimization goal is. This is the default value.
  - **Default** - FireDAC does not optimize anything. Previous mode.
  - **All** - FireDAC always optimizes resultset fetching to finish fetching all records.
  - **First** - FireDAC always optimizes resultset fetching to finish fetching the first record.
- Better High-DPI scaling for TFDQuery Editor Options and for the font in the query editor.
- Added a proper version of the MongoDB client library to provide support for logging into a MongoDB server and for a new transaction feature
- Made TBindingExpression case insensitive.
- Verified FireDAC drivers' compatibility with PostgreSQL version 14 and version 15.1.
- Added new NCharReplacement configuration for FireDAC Oracle connection and added support for q['...'] "NCHAR String Literal Replacement".

HTTP and Rest Client libraries

For HTTP and Rest Client libraries, RAD Studio 11.3 offers the following enhancements.

- The TMultipartContentParser property avoids treating every part with ContentType as File.
- Add automatic token refresh to TOAuth2Authenticator, to improve support of Google REST services authentication and the expiration of TOAuth2Authenticator access tokens.
- Better support for NestedElements in TRESTResponseDataSetAdapter.
- Added support for using client certificates with REST components on Android.
- RAD Server, the TEMSDataSetResource component LIST functionality now returns properly formatted dates in the JSON structure.
- Addressed a memory leak in a SOAP Server application when serving an attachment.

Other Enhancements

RAD Studio 11.3 offers a new notarization process for macOS that Apple introduced (https://developer.apple.com/documentation/security/notarizing_macos_software_before_distribution), starting with XCode 14. Also, Windows signing now includes support for the time stamping of MSIX files.

See Also

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