



embarcadero

MAY 2017 ROADMAP

& Commentary from Product Management

May 2017 Roadmap & Commentary from Product Management

After the 10.2 Tokyo release that introduced Delphi for Linux, we launched our annual developer survey. The survey ran for a couple of weeks and ended at the end of April. Participation in our survey was exceptional (50% higher participation than in the past), providing lots of invaluable feedback to the product management team. We'd like to thank everyone who took the time to participate in our survey. The survey results were very helpful for finalizing our roadmap plans.

This document contains the May 2017 Roadmap for RAD Studio, Delphi and C++Builder along with commentary by product management. It is all really exciting to everyone here at Embarcadero. In here you will find the key features that we planned for calendar year 2017/2018. The plans are aligned with Embarcadero's fiscal year which starts April 2017 and ends in March 2018. We generally plan to update our roadmap every 6 months.

Safe Harbor Statement

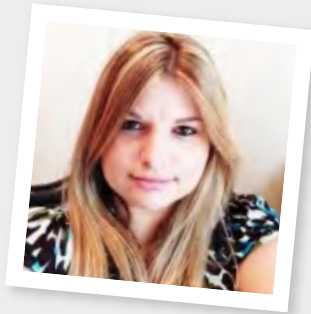
These plans and roadmap represent our intentions as of this date, but our development plans and priorities are subject to change.

Accordingly, we can't offer any commitments or other forms of assurance that we'll ultimately release any or all of the described products on the schedule or in the order described, or at all.

These general indications of development schedules or "product roadmaps" should not be interpreted or construed as any form of a commitment, and our customers' rights to upgrades, updates, enhancements and other maintenance releases will be set forth only in the applicable software license agreement.

Meet the Product Management Team

Everyone on the product management team works together. They each also have areas of focus. You will find their comments mixed into the roadmap information.



Sarina DuPont Senior Product Manager
sarina.dupont@embarcadero.com

Sarina focuses on FireMonkey, the components libraries UX and styles, RAD Server, the installation experience, plus demos and documentation



Marco Cantu Delphi Product Manager
marco.cantu@embarcadero.com

Marco focuses on the Delphi language and RTL, the VCL library and Windows integration, all things database and Web oriented



David Millington C++ Product Manager
david.millington@embarcadero.com

David focuses on the C++ language, the debugger, and IDE and user experience.

RAD Studio Today

Key features delivered in recent months:

Delphi Linux compiler and RTL, along with complete toolchain for Linux server development, deployment and debugging

Windows 10 Store support via Desktop Bridge

New VCL controls and QuickEdit designers

C++ compiler and linker improvements

RAD Studio Personas

Focus Areas for CY 2017 / 2018



Windows Desktop Developers

IDE and VCL Enhancements

Delphi Language Features

Update to the latest C++ language standards

Continued Windows 10 platform support



Mobile Developers

Support latest versions of iOS and Android Multi-device support for Windows and macOS

Android native platform controls Platform styling improvements



Enterprise Developers

Linux server support for C++ and additional Linux features REST API Microservices support with RAD Server enhancements

DataSnap enhancements

Support for the latest database drivers



www.embarcadero.com

RAD Studio CY 2017 / CY 2018 Timeline

Features are not committed until completed and released with general availability

2017

2018

10.2.1 Tokyo

10.2.2 Tokyo

10.3 "Carnival"



RAD Studio (all IDEs)

- VCL improvements (HiDPI)



RAD Studio (all IDEs)

- Android Z-Order (native windowing) support
- Dark IDE theme
- IDE UI/UX improvements
- Additional VCL GUI controls



RAD Studio (all IDEs)

- Native Android controls in FireMonkey
- IDE UI/UX improvements
- Consolidate debuggers
- Simplified GetIt install experience



Delphi

- Additional Delphi Linux features not in 10.2



Delphi

- Windows 10/WinRT/VCL updates



Delphi

- Language Enhancements
- 64-bit macOS platform support



C++ Builder

- Linux server support



C++ Builder

- Language standards: CMake, C++ library integration
- C++ rename refactoring / code completion



C++ Builder

- Language standards: Clang support, C++ library integration (cont.)

RAD Studio 10.2.1 Tokyo - 2017

Enabling Enterprise developers to target Linux servers with C++ and Delphi

RX

RAD Studio (all IDEs)

- VCL improvements (HiDPI)
- Support for Windows 10 Creators Update
- Quality and performance enhancements for all subsystems, specifically Android platform

CX

C++ Builder

- 64-bit Linux server support
- Clang-enhanced C++ compiler and linker for Linux 64-bit
- IDE integration, debugging support
- Access to the Delphi RTL plus C++ specific libraries like Boost

DX

Delphi

- Additional Delphi Linux features not in 10.2: further RTL libraries, improved debugging experience, and fixes of discovered issues

RAD Studio 10.2.2 Tokyo - 2017

Enabling Windows and mobile developers with new platform specific features

RX

RAD Studio (all IDEs)

- Android Z-Order (native windowing) support, the foundation for platform controls
- Dark IDE theme
- IDE UI/UX improvements in several keyareas
- Additional VCL GUI controls, completing our set of WinRT/Windows 10 UI controls
- Quality and performance enhancements
- Support for new versions of mobile and desktop operating systems shipping in 2017

CX

C++ Builder

- More focus on language standards with CMake and better C++ libraries integration
- C++ rename refactoring support, or improved code completion and code insight

DX

Delphi

- Windows 10/WinRT/VCL updates, including more extensive WinRT APIs and platform APIs coverage for Windows



RAD Studio 10.3 "Carnival" - 2018

Native 64-bit macOS platform toolchain, Delphi and C++ language modernization on all platforms

RX

RAD Studio (all IDEs)

- Native Android controls in FireMonkey, starting with input controls
- Further IDE UI/UX improvements
- Consolidate multiple debuggers, focusing on the LLDB technology
- Simplified GetIt install experience

CX

C++ Builder

- Improved support for the latest C++ language standards, and additional C++ library integration work

DX

Delphi

- 64-bit macOS platform support, including the entire development, deploying, and debugging experience
- Language Enhancements with support for Nullable Types and additional syntax improvements



RAD Server

REST API micro-services support, for mobile, desktop, and web architectures

Features expected in CY 2017 / CY 2018 timeframe

- Enterprise integration and security improvements
 - Active Directory and Kerberos
- Support for Angular.JS clients
- Integration with Enterprise REST Services (Salesforce/MSFT Dynamics)
- RAD Server Industry Templates



Research Areas**

Getting ready for future platforms and technologies

Platforms

- BitCode support and other LLVM-related improvements
- Target ARM IoT platforms (Raspberry PI, Windows IoT)
- Delphi ARC model on Windows

Under Consideration

- Linux GUI support
- Further FMX native controls (additional controls and platforms)
- Unified installation experience with combined Web and offline installer
- 64-bit tooling and/or IDE
- Tools for building Web clients

*** We don't know if and when these technologies will be delivered, but the PM and R&D teams are investigating these platforms and technologies for future consideration and adoption in RAD Studio*

Commentary from Sarina DuPont

Senior Product Manager

sarina.dupont@embarcadero.com



FireMonkey Native Controls Support on Android

We currently support pixel perfect, bitmap based control rendering for all of our UI controls across all four supported platforms, in addition to native control rendering for key text input and list controls on iOS and Windows. As part of our roadmap, we plan on expanding our native user interface control support to Android.

We are looking to do this in stages, starting with Z-Order support, followed by native controls on Android, focusing on the same key text input and list controls that we have native presentation support for today on iOS and Windows. This provides a lot of flexibility as it allows developers to choose between Styled and Platform control types on a per control basis for controls that include both control types, while still being able to leverage custom FireMonkey styles in multi-device applications.

We plan to introduce additional templates and design elements to aid the design process for FMX applications. We will collaborate with our component vendors in this space to create a richer portfolio of styles and features to make FMX the perfect choice for fast application development.



RAD Server Enhancements

If you're not familiar with RAD Server yet, I'd encourage you to have a look. RAD Server is our turn-key application foundation for rapidly building and deploying services based applications (self-hosted PaaS).

RAD Server enables developers to quickly build new application backends or migrate existing Delphi or C++ client/server business logic to a modern service based architecture that is open, stateless, secure and scalable. In the 10.2 Tokyo release, we added multi-tenancy support to RAD Server and the ability to deploy your solution to both Windows and Linux servers. This is a game changer for many of our ISV customers who can now create more flexible and efficient deployments for their customer base.

We have plans to deliver Active Directory support for RAD Server and AD synchronization for accounts and Kerberos authentication for clients along with RAD Server support for AngularJS clients. Something else I am really excited about are the RAD Server project templates for several key industries that we are working on. Industry templates are designed to provide you with a great starting point for building your own RAD Server solution. Stay tuned for more info on that in the future.



www.embarcadero.com

Commentary from Marco Cantu

Delphi Product Manager

marco.cantu@embarcadero.com



Delphi Language: New Features and macOS Focus

Over the next 12 months we are planning to release a new round of features for the Delphi language, particularly focused around the addition of nullable types to the type system, but also improvements on records initialization and finalization. We are also evaluating some “syntax sugar” enhancements, that is smaller improvements not adding much power to the language but allowing developers to write less or cleaner code.

The other area of development will be around the macOS compiler, with the goal of moving to native 64bit support for that platform (most likely leveraging shared technology with the recently released Linux 64bit compiler).

Code optimization is also very dear to our customers and will be another focus area for the Delphi compiler.



VCL and Windows

The VCL library has been growing and it is finding a place in the modern Windows 10 ecosystem, thanks to styling, new WinRT-like controls, and platform integration including the support for the Windows Desktop Bridge. Over the next year, the plan is to continue in that

same direction, exposing more native WinRT APIs, providing additional WinRT-like VCL controls, improving VCL support for modern hardware (like HighDPI screens), and keeping up with the platform evolution in terms of APIs and user experience.



All Things Database and Web Oriented

In the database area, the official roadmap doesn't mention the work we are expecting to do to keep FireDAC drivers up to date with the various RDBMS, since that along with consolidation and improvements to the core technology is part of the expected work. We are planning maintenance updates to the recent HTML and REST client libraries, along with significant fixes around DataSnap. Most new development is likely to appear in RAD Server (EMS), but we are not forgetting our core infrastructure -- which was just recently ported to Linux in the 10.2 Tokyo release.

We have some exciting plans to introduce new Enterprise Application connectivity adapters that will make connectivity from RAD Studio to many applications, such as Salesforce and Netsuite, as easy as working with the existing FireDAC technology. This will give our users unique accelerators for even faster and broader application development.

Further we are working to expand web development options that are fast and robust, so stay tuned.

Commentary from David Millington

C++ Product Manager

david.millington@embarcadero.com



C++ Builder

C++ Builder has some very unique and powerful tooling, especially through the VCL and FMX frameworks, which are better in many areas than other options in the market. Our language extensions further differentiate it, even compared to strong products such as Visual Studio.

The first key new capability we plan to deliver this year is C++ for Linux Server. This will make Linux deployments for new and existing server or console apps seamless.



Other key focus areas for C++ Builder include:

Bring our language support to C++17. We are addressing this by upgrading the version of Clang we extend, and plan to keep this up to date as Clang evolves.

Make it easy to work with the larger C++ ecosystem. Here, we want to add support for CMake, both to recognize our compilers and some level of in-IDE support. We also want to ensure that a number of popular C++ libraries can be easily brought into projects and used with C++ Builder.



Improving the IDE tooling

This addresses areas such as code completion, refactoring, and so forth. 10.2 had some fantastic improvements in compiler quality, and you can expect ongoing work. This also includes areas such as Delphi / C++ compatibility, where we have a focus on ensuring the interaction layer works as a C++ developer would expect.

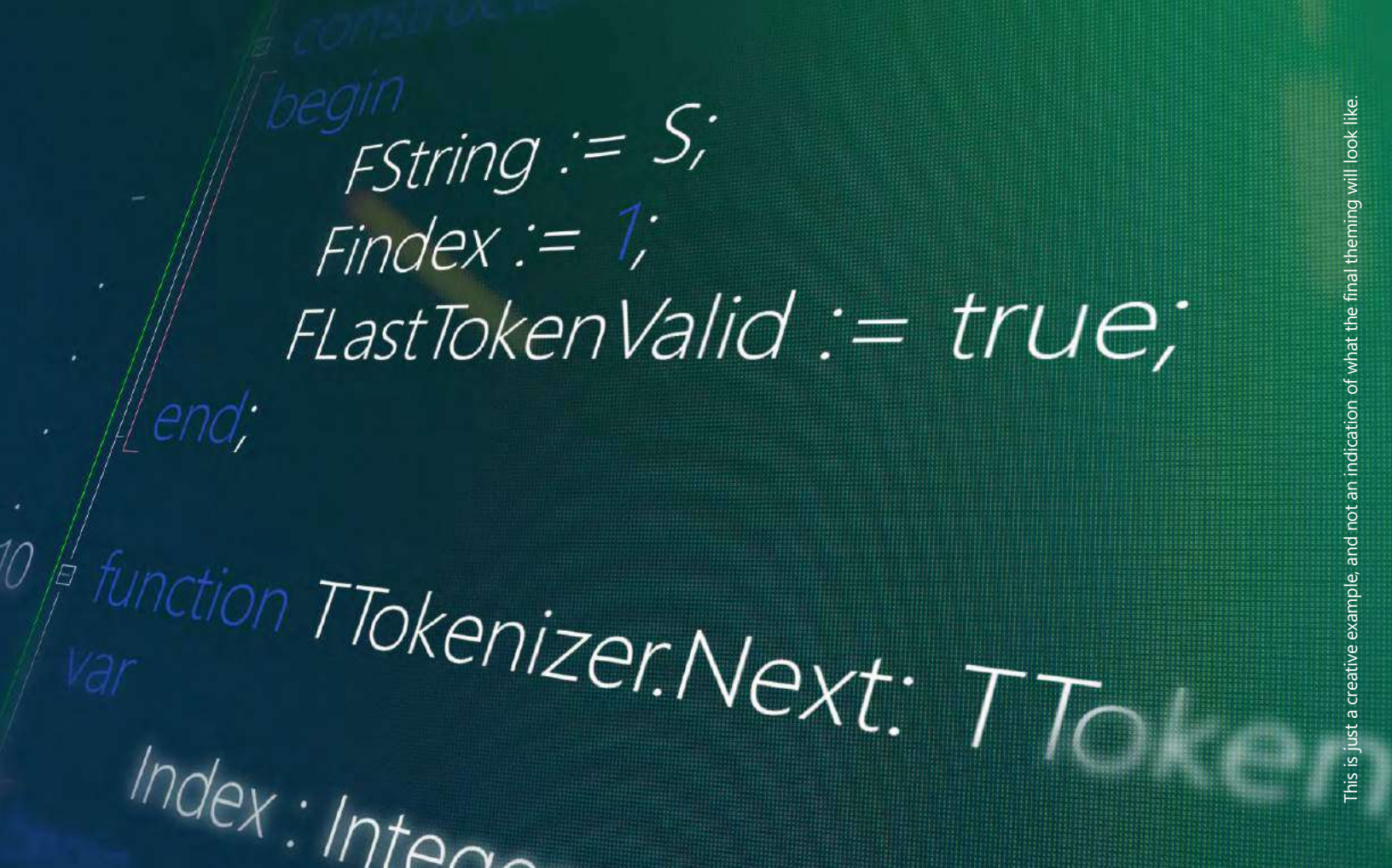
You may note that macOS 64 is listed for Delphi, not C++. Don't worry: this does not mean that we are not planning macOS 64 support for C++. We are considering whether to implement with our existing or new upgraded C++ toolchains, and Delphi's support will enable us to deliver C++ support afterwards.



Debugger

In 10.2, we delivered several key quality improvements addressing customer issues, especially on Win64, iOS and Android, and that will continue. We also plan quality improvements for Linux Server.

We have several different debugger backends on various platforms, and as an ongoing process are working towards consolidating them towards one, LLDB. This provides a number of benefits, not least of which is that a single debugger codebase will result in debugger features and bug fixes being available on multiple platforms much more quickly.



This is just a creative example, and not an indication of what the final theming will look like.

IDE

The IDE is one of the most important parts of RAD Studio, because it is where all other parts of it are presented and through which are interacted with. One personal drive of mine is to make significant quality and feature improvements here and you can look forward to some great things over the next year.

One example is adding a dark theme, something many users have requested. Many tools, including Photoshop and other IDEs, have a 'dark' theme for nighttime use. Ours will be easily toggleable on and off.

Other areas we are looking at include component icons, with an eye towards consistent styling, clarity, and recognizability; overall IDE tidiness and layout; overall UX, both for ease of use and ease of seeing data; quality, for code completion and related items; and new features.