RAD STUDIO 13 PRODUCT FEATURE MATRIX

Each feature in the following table has an indication if it is available in Delphi and/or C++Builder. RAD Studio editions include both languages. Each column covers multiple, similar editions, with any difference specifically called out. Some of the features are only available for download from the IDE in the GetIt Package Manager, requiring an Internet connection and an active Update Subscription. Community Edition[1] is currently available only for version 12.1, has a limited license, and doesn't include the bonus features available in the GetIt package manager.

TABLE OF CONTENTS

INTEGRATED COMPILERS AND TOOLCHAINS LICENSING AND TOOL MANAGEABILITY	APPLICATION PLATFORMS, INTEGRATED FRAMEWORKS, FMX APPLICATION PLATFORM (FIREMONKEY)
VISUAL COMPONENT LIBRARY (VCL)	COM/DCOM SUPPORT
AI-POWERED FEATURES	VISUAL LIVEBINDINGS
INTEGRATED BUILD TOOLS AND PROJECT SUPPORT	RECENT C++ LANGUAGE AND CORE RTL FEATURES
RECENT DELPHI LANGUAGE AND CORE RTL FEATURES	SHARED RUNTIME LIBRARY FEATURES
INTEGRATED DEVELOPMENT ENVIRONMENT	SOURCE CODE MANAGEMENT
REFACTORINGS	CODE SEARCH AND NAVIGATION
INTEGRATED DEBUGGING	INTEGRATED HELP
INTEGRATED UNIT TESTING	DATABASE APPLICATION DEVELOPMENT
FIREDAC MULTI-DEVICE DATA ACCESS LIBRARY	OTHER DATABASE TECHNOLOGIES
WEB BROKER AND WEBSTENCILS	RAD SERVER
DATASNAP MULTI-TIER	INCLUDED DATABASE TOOLS AND DATABASE ENGINES
CLOUD SUPPORT	REST CLIENT LIBRARY AND BAAS
XML AND SOAP	FILE COMPARE WITH BEYOND COMPARE TEXT COMPARE
FMXLINUX	LOGGING WITH CODESITE EXPRESS
ADDITIONAL COMPONENTS	SENCHA EXT JS PROFESSIONAL EDITION
AQUA DATA STUDIO	

		PROFESSIONAL COMMUNITY[1]	ARCHITECT ENTERPRISE
INTEGRATED COMPILERS AND TOOLCHAINS			
ENHANCED IN 13 FLORENCE	Delphi 32-bit compiler and toolchain for Windows (dcc32) Delphi 64-bit compiler and toolchain for Windows (dcc64)	✓	✓
ENHANCED IN 13 FLORENCE	64-bit binary versions of the Delphi 32-bit (dcc32) and Delphi 64-bit (dcc64) command line compilers, offering a very large memory address space	✓	✓
ENHANCED IN 13 FLORENCE	Delphi 64-bit compiler and toolchain for Linux Intel (dcclinux64)		✓
ENHANCED IN 13 FLORENCE	Delphi 64-bit compiler and toolchain for macOS Intel (dccosx64)	✓	✓
ENHANCED IN 13 FLORENCE	Delphi 64-bit compiler and toolchain for macOS ARM (dccosxarm64)	✓	✓
ENHANCED IN 13 FLORENCE	Delphi 64-bit compiler and toolchain for iOS devices (dcciosarm64)	✓	✓
ENHANCED IN 13 FLORENCE	Delphi 64-bit compiler and toolchain for iOS Simulator on ARM 64-bit macOS (dcciossimarm64)	✓	✓
ENHANCED IN 13 FLORENCE	Delphi 32-bit compiler and toolchain for Android ARM devices (dccaarm)	✓	✓
ENHANCED IN 13 FLORENCE	Delphi 64-bit ARM compiler and toolchain for Android 64-bit platform (dccaarm64)	✓	✓
	C++ 17 32-bit Clang-enhanced compiler and toolchain for Windows (bcc32x and bcc32c).	✓	✓
	Classic Embarcadero C++ compiler and toolchain for Windows (bcc32)	✓	✓
	C++ 17 64-bit Clang-enhanced compiler and toolchain for Windows (bcc64)	✓	✓

[1] Community Edition is currently available for the 12.1 Athens release, which doesn't include all of the features in 13.

NEW IN 13 FLORENCE	C++ compiler for Win64, based on Clang 20, UCRT for the C runtime, a mix of LLVM and MinGW for the C++ runtime, and LLVM's libc++ STL. The new toolchain is fully integrated in the product (bcc64x). New version updated from Clang 15.		

LICENSING

	Licensing suitable for larger teams with no restriction on the number of licenses you can purchase or use and on revenue derived from the product (see EULA for details)	PROFESSIONAL only	
	Network named or concurrent licenses available	PROFESSIONAL only	
ENHANCED IN 13 FLORENCE	Earlier version licenses included in network licenses (Delphi 2007–12, C++ Builder 2007–12). Delphi 7 and C++Builder 6 available separately	PROFESSIONAL only	
ENHANCED IN 13 FLORENCE	Access to licenses and downloads for earlier versions with purchase of standalone product licenses (Delphi 2007–12, C++Builder 2007–12, Delphi 7, and C++Builder 6)	PROFESSIONAL only	

APPLICATION PLATFORMS, INTEGRATED FRAMEWORKS, DESIGNERS AND SDKS

ENHANCED IN 13 FLORENCE	FireMonkey Application Platform for creating 32-bit Windows applications and 64-bit Windows applications for Windows 11, Windows 10; Windows Server 2025 and 2022		
ENHANCED IN 13 FLORENCE	FireMonkey Application Platform for creating 64-bit macOS applications for Intel and ARM CPUs, that run on macOS 15 Sequoia and macOS 14 Sonoma, including support for universal binary		
ENHANCED IN 13 FLORENCE	FireMonkey Application Platform for creating iOS ARM 64-bit applications for iOS 18		
ENHANCED IN 12 ATHENS	Delphi iOS Simulator support for ARM 64-bit macOS		
ENHANCED IN 13 FLORENCE	FireMonkey Application Platform for creating native 32-bit and native 64-bit Android ARM applications for Android 16, 15, and 14. Supports Android API level 35. ** Android development requires the Android SDK and NDK from Google, which cannot be fully installed without an Internet connection*		
ENHANCED IN 13 FLORENCE	VCL (Visual Component Library) for rapidly building 32-bit and 64-bit applications for Windows 11, Windows 10; Windows Server 2025 and 2022		
ENHANCED IN 11.3 ALEXANDRIA	Windows MSIX package format deployment (providing support for deploying to the Microsoft Store), including the management of additional files, MSIX packaging and code signing automatically invoking the platform SDK. The Windows signing support in the IDE since 11.3 includes support for the time stamping of MSIX files.		
ENHANCED IN 13 FLORENCE	Microsoft Windows SDK support for Windows 11, 10 and 7 APIs. Includes support for calling both classic Win32/Win64 APIs and enhanced support for WinRT APIs; required for the Windows App SDK (aka Project Reunion)		

FIREMONKEY APPLICATION PLATFORM

ENHANCED IN 11 ALEXANDRIA	FireUI multi-device designer for visually building mobile and desktop applications and customize them at design time for different form factors and operating systems (limited to Windows for C++)		
	FireUI multi-device designer ability to make changes on the main user interface that are propagated to other device specific user interfaces and fine tune a device specific form factor without affecting other views; support for adding additional custom views to the designer		
	FireUI App Previews lets you preview your FireMonkey application on iOS, Android, desktop targets as you are designing the app. Changes made in the FMX Designer are broadcasted in real time to connected devices using the appropriate Styles for the devices.		

	Several dozen UI controls for building FMX applications, covering the most common UI guidelines and including also 3D support, for rapidly building multi-device applications		
ENHANCED IN 13 FLORENCE	FireMonkey Skia support including direct APIs, specific UI controls, and the automatic mapping of FireMonkey UI controls UI rendering via Skia. In general terms, Skia improves the quality of the rendering and the performance. Skia4Delphi has been updated in 13.		
INTRODUCED IN 12 ATHENS	FireMonkey Skia-based controls: TSkAnimatedImage, TSkLabel, TSkPaintBox, TSkAnimatedPaintBox, and TSkSvg		
INTRODUCED IN 12 ATHENS	Exclusive Skia features not found in the Skia4Delphi library: • Vulkan Backend support for Android (Delphi only) • Skia Shading Language (SKSL) • Animated Codec: WebP Encoder • Printer Support and Printing to PDF		
NEW IN 13 FLORENCE	New Display Link Service offering highly improved stability and smoothness of animations		
ENHANCED IN 11.1 ALEXANDRIA	FireMonkey for Windows uses the same DP model (rather than Pixel model) of all platforms, offering significant enhancement for the apps rendering on Windows HighDPI and 4K monitors		
	Metal Drivers GPU Support for macOS and iOS		
INTRODUCED IN 11.3 ALEXANDRIA	Mobile Biometric Authentication component, called TBiometricAuth, which provides a means of authorization to use an application or some specific features via Biometrics (e.g., Face ID on iOS devices or Fingerprint on Android devices).		
NEW IN 13 FLORENCE	New MaskEdit component		
NEW IN 13 FLORENCE	New IFMXSpellCheckerServiceEx interface		
NEW IN 13 FLORENCE	New ApplicationEvents component		
INTRODUCED IN 12 ATHENS	Styled Memo Implementation with enhanced IME support, available for all supported platforms		
INTRODUCED IN 12 ATHENS	Since version 12, FireMonkey TEdit supports Undo functionality, using the recently added TUndoManager is shared between TEdit and TMemo		
INTRODUCED IN 12 ATHENS	Editing improvements with Virtual Keyboard for iOS, better interactive text selection, caret point management and finger gestures on Android		
	TBufferedLayout Component for faster rendering of child controls		
	Support for iOS Launch Storyboard		
INTRODUCED IN 12 ATHENS	Split screen app support (side-by-side apps) for both iOS and Android		
INTRODUCED IN 12 ATHENS	Android's adaptive icons		
	Automatic iOS Theme check and theme change notification		
	Address Book component for iOS and Android, allowing access to and extensive modification of device contacts and groups and including monitoring of address book changes by other applications		
	Google Firebase Push Notification support for the Android platform, with IDE-integrated configuration and updated Google Play services		
	QuickEdit designers for FireMonkey, including a pane for editing common control properties faster and several other additions to the IDE designer experience		
	Extended clipboard service support, with IFMXExtendedClipboardService interface, with support for querying and working with text, images, and custom formats		
	Support for accelerator keys on Windows, compatible with other platforms and implemented using the IFMXAcceleratorKeyRegistryService platform service. There is also a FocusControl property for the Label control to specify the focus target for the label's accelerator		
	Grid controls improvements including support for ImageList, AutoDisplacement and CancelEditingByDefault options, and StringGrid supports for glyph and currency columns		

	ListView item designer, providing an interactive way to work with ListView items; Touch animation support for TListView		
	Enhanced TFont and TFontStyle to support multi-weight, multi-style fonts		
	Improvements to MessageBox, ShowMessage and InputQuery offering both synchronous (blocking) implementation and asynchronous (non-blocking) implementation		
	Mouse-over Hints support for FireMonkey visual controls on desktop		
	FireMonkey zOrder support for Windows and native style presentation for Windows for Edit and Memo platform controls		
	FireMonkey zOrder support for Android and native style presentation for Android for Edit, Calendar, Switch, and Multiview controls		
	Touch animation for Android platform		
	FireMonkey apps can receive Android intents, regardless of the source (email, web link, other app)		
	FireMonkey Behavior service for automatically adjusting UI properties to the target operating system standards		
	MultiView component to create a UI that adapts to the form factor, from mobile to tablet, from portrait to landscape. Includes specific support for Windows 10 UI, options to manage borders appearance, and alignments for docked panel mode		
	Multi-Threading support for TBitmap, TCanvas and TContext3D		
	Unification of Delphi and Java threads on Android		
	Radiant Shapes controls collection* ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	Specific TSwitch behavior for Windows 10 UI		
	Edit and Calendar ability to dynamically render a platform native UI control on iOS; Memo platform controls for iOS for highest input fidelity		
	TGrid, TListView, TCalendar, TSwitch and TScrollBox platform controls for iOS		
	MapView component for mobile platforms (iOS and Android) supporting platform specific interactive mapping libraries (respectively, by Apple and Google)		
ENHANCED IN 13 FLORENCE	WebBrowser component for Windows, with support for both the classic IE ActiveX and the Microsoft WebView 2 control (Chromium-based Edge). Updated in version 13.		
	WebBrowser component for macOS based on WKWebView platform API		
	Improved mixing of platform and styled components, with platform styling options for iOS and controls Z-order improvements (with ControlType property for non-platform controls)		
	ImageList component for FireMonkey, for storing and manipulating multiple images and sub-images in a single component; several components can pick their graphical elements from the ImageList, rather than a standalone image		
	Multi-monitor support for desktop platforms		
	Splash screen support for Android, with specific project options matching the iOS ones		
	Form-level multi-touch support for all available platforms		
	Tint support for buttons and toolbars on iOS and Android (for control surface and controls icons)		
ENHANCED IN 11 ALEXANDRIA	In-app purchase component for iOS and Android, to help monetize your applications. Android solution Google Play Billing Library Version 4		
	Advertising component for Android, including AdMob support based on the Google Mobile Ads SDK (updated to version 7.0 in RAD Studio 10.2.3), also to help you monetize your mobile applications		
	Advertising component for iOS, including AdMob support based on the Google Firebase SDK		
	Push notifications support for mobile platforms, to receive remote notifications even if the app is not running (including Firebase support for Android and iOS clients)		

	FireMonkey support for AppTethering across desktop and mobile applications, including both Wifi and Bluetooth support, with protocol enhancements, including encryption hooks and events to handle when a profile is connected or disconnected		
	Native message alerts for iOS and Android		
	Custom Pickers for iOS and Android, including Date Picker (DateEdit component) and Time Picker, Time Picker for desktop platforms		
ENHANCED IN 11.3 ALEXANDRIA	Phone Dialer Support for iOS and Android. Enhanced in 11.3 with IFMXPhoneDialerListenerService, implemented for Android. The service is intended for tracking changes in the state of the phone system service (Calls, Carrier, CallState).		
	Built-in text editing mode for Edit and Memo controls on Android and iOS, including Cut/Copy/Paste/Zoom; Spell Check Support for text input controls on iOS and Android		
	Notification Center Component for using notifications in your iOS and Android applications, including notification sound and supporting also Windows 10 Notifications		
	Gesture support for iOS and Android including swipe (pan), tap, tap and hold, double tap and pinch & zoom		
	FireMonkey Mobile Application Wizard with header/footer, tabbed, and master detail templates		
	Full set of native iOS and Android styling options for buttons, toolbars, list boxes, sliders, switch controls		
	Swipe-to-Delete and Pull-to-Refresh on iOS and Android with the ListView component; search filtering for ListView		
	WebBrowser component for loading and displaying web content in your iOS and Android applications (with corresponding desktop component)		
	Extended ListBox capabilities for iOS and Android, including header/footer/grouped styling, embedded search and expanded LiveBindings support		
	Support for various keyboard types on iOS and Android		
	TMagnifier component for easily zooming in on images and text		
	TCameraComponent component for accessing the front and back camera, the flash, and more device camera features		
	TabControl with settings for using tabs as views and a PlatformDefault tab position to automatically adapt tab layout to the current mobile platform, support for TabControl tab icons on Android (Delphi only), slide transition action for TTabControl		
	FireMonkey Frames for designing portions of the UI and repeating them on multiple forms		
	Toggle Device Skin to show/hide mobile device background in the multi-device designer		
	FireMonkey Save State feature for persisting the forms status between sessions (and when the program is stopped by the operating system)		
	macOS Full screen Mode		
	Use of IFMXDragDropService to drag data to another application on macOS		
	Actions and ActionLists for FireMonkey for better separation of the application logic from the user interface		
	Ready-to-use action for accessing the camera application and for accessing images from the camera roll on iOS and Android		
	Share Sheet support on iOS and Android for sharing content via Messaging, Mail, Facebook, Twitter etc.		
	FireMonkey control anchors for flexible positioning and sizing of the visual controls; Layout Managers (Flow Layout, Grid Layout) for flexible child controls positioning and sizing		
	Location Services component for GPS or triangulation, including geocoding support		
	Orientation sensor (gyroscope/compass) on iOS, Android, and on Windows (requires compatible hardware)		
	Motion Sensor component for accelerometer access on iOS, Android, and on Windows (requires compatible hardware)		

	3D File Import Support		
	Non-Client Area Application Styling for Windows and Mac (Delphi only)		
	Support for Windows gestures. On-screen keyboard support for Windows touch devices		
	Edit control with styleable glyph with built-in style support for search bar and password control		
	Animated progress glyphs, including Windows ring		
	StyleLookUp support with preview of available designs for the selected component		
	Shared external redistributable bitmap style designer for VCL and FMX		
	Premium FireMonkey multi-device styles* with theming support for all supported platforms ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	Bitmap Style Designer enhancement, including templates for Windows, Mac OS and Android styles, ability to create blank styles for different platforms, and to adjust objects for different graphic resolutions. Some platforms are Delphi only.		
	Significantly improved integrated FireMonkey Style Designer, and enhancement to the TStyleBook component to support a collection of styles for various platforms		
	Bitmap Links editor, to modifying TBitmapLink properties directly inside the IDE		
	Native UI and custom UI control styles		
	FireMonkey Visual Form Inheritance		
	Use one shader language for all rendering libraries (OpenGL or 3D)		

VISUAL COMPONENT LIBRARY (VCL)

	Object-oriented, fully extensible and reusable visual component library (VCL) and applications architecture for Windows development		
NEW IN 13 FLORENCE	Integration of the UIAutomation Microsoft interfaces, which can be used to support accessibility and to build automatic UI testing		
ENHANCED IN 11 ALEXANDRIA	VCL Form Designer to visually build native Windows applications, with live snap-to hints and layout guidelines		
	Visual Form Inheritance and Form linking to reduce coding and simplify maintenance; Frames for building and reusing compound components		
	Object Repository for storing and reusing forms and Data Modules		
	Object Inspector to set component properties and events		
INTRODUCED IN 12 ATHENS	VCL Skia-based controls: TSkAnimatedImage, TSkLabel, TSkPaintBox, TSkAnimatedPaintBox, and TSkSvg		
INTRODUCED IN 12 ATHENS	Support for full HighDPI and VCL styles in MDI applications. The MDI support in VCL has been reworked and expanded to overcome Windows platform limitations. Includes an updated MDI form Wizard		
ENHANCED IN 13 FLORENCE	TFormTabsBar control and IFormVisualManager interface to help build general tab-based UIs and a MDI tabbed forms architecture		
	VCL Styles support for High-DPI and 4K monitors, with support for multiple resolution of graphical elements, automatic scaling, and improved element sizing on different DPI configurations		
	A dozen of ready to use High-DPI optimized VCL styles		
	Per-control VCL styling support		
INTRODUCED IN 12 ATHENS	Extended TFont VCL class with DPI-related and improved scaling support		
ENHANCED IN 13 FLORENCE	Custom Title Bars and the TitleBarPanel Component, enhanced in 13 with support for TitleBar styling		
	TVirtualImage component with support for images with multiple resolutions		

ENHANCED IN 13 FLORENCE	TEdgeBrowser Component integrating Microsoft's Chromium-based Edge WebView2 component and changes to the TWebBrowser component to support both IE and Edge-- based on Microsoft release version of Edge WebView2 component and supporting custom cache folders, with support for the UserDataFolder and the ExecutableFolder configurations. Enhanced in 13.0 with additional properties and methods, mapped to features in recent versions of the WebView 2 control		
ENHANCED IN 13 FLORENCE	VCL TControlList control, a flexible and virtualized list control, designed as a high-performance control for very long lists and a modern look and feel, with support for the selection of multiple items and other enhancements.		
ENHANCED IN 12.1 ATHENS	VCL TNumberBox control, a modern-looking numeric input control, for integer, floating point numbers, and currency values. Enhanced in 12.1 with better support for negative values and scientific notation		
	QuickEdit designers for VCL, including additional items in the designer menu, a pane for editing common control properties faster, and several other additions to the IDE designer experience		
INTRODUCED IN 12 ATHENS	Additional property editors: string list editor and multiline string editor (originally part of KSVC, Konopka Signature VCL Controls)		
INTRODUCED IN 12 ATHENS	Quick configuration dialogs for buttons, radio groups and panels (originally part of KSVC, Konopka Signature VCL Controls)		
	Windows components fully integrated with the classic Windows API and the modern WinRT API		
ENHANCED IN 13 FLORENCE	Set of units with the complete Windows API headers converted to Object Pascal. This API conversion includes 311 Delphi header files* with 41 MB of code, covering the entire platform API provided by Microsoft. ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	Complete VCL library source code to help writing custom components, debug your applications, and fully understand the library behavior	Limited in COMMUNITY	
ENHANCED IN 12.3 ATHENS	TImageCollection and DPI-aware TVirtualImageList components to manage multiple-resolution images and improve application support for high-DPI monitors and application form scaling. Includes design time support with an Image Collection editor and a Virtual Image List editor.		
	Support for Windows "Per Monitor V2" mode in application manifest and enhanced system metrics API support (requires Windows 10 Creator's Update, build 1703)		
	High-DPI Awareness and 4K monitor support, plus Windows 8.1/10 multi-monitor support for VCL applications		
	Custom VCL controls corresponding to some of the Windows 10 UI controls, which can also be used on previous version of Windows		
	DatePicker and TimePicker VCL controls; StackPanel and CardPanel VCL controls		
	ActivityIndicator, SearchBox, RelativePanel, ToggleSwitch, SplitView VCL control; modern looking SelectDirectory function		
	CalendarView and CalendarPicker VCL controls		
	VCL Font Scaling improvements for higher DPI and higher resolution monitors		
	Updated Konopka Signature VCL Controls (over 200 additional Windows UI controls)* ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	VCL sensor components for Windows tablets		
	VCL support for AppTethering across desktop and mobile FireMonkey applications written in Delphi, with Wifi and Bluetooth support with extensions and protocol enhancements, including encryption hooks, including events to handle when a profile is connected or disconnected		
	Taskbar component with progress bars, overlay icons, custom buttons, and multiple previews; JumpList component for adding custom menus to taskbar buttons		
	VCL Styles: Create VCL applications with enhanced GUI, with additional Windows 10 Styles for VCL and including Windows touch tablets optimized styles; support for styling of main menus, popup menus, and system menus		
	Windows 10 specific VCL styles, to build applications matching Microsoft's Modern look and feel; Premium styles including TabletLight, Sky, and Glow VCL styles		

	10 Premium VCL Windows styles* that users can apply to their Windows applications ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	Create your own custom styles with the VCL Style Designer		
	VCL Styling improvements, including support for styling common dialogs and the TWebBrowser component		
INTRODUCED IN 11 ALEXANDRIA	TRichEdit Component updated to RichEdit 4.1 (MSFTEDIT.dll) with support for transparency, URL detection, spell checking, text attributes and more		
ENHANCED IN 11 ALEXANDRIA	Small enhancements to TMemo, TDBMemo, TGroupBox, TComboBox, TCoolBar, TNumberBox and other components.		
INTRODUCED IN 11 ALEXANDRIA	TTreeView support for CheckBoxes		
INTRODUCED IN 11 ALEXANDRIA	TLabelDBEdit component, a data-aware version of the TLabelEdit component		
	Integrated gesturing framework with multi-OS support for Windows 8. Gesture Designer enables full control of all gesture design and interaction; Gesture components (TGestureListView, TGesturePreview, TGestureRecorder) to build gesture creation and management into VCL based applications; Interactive multi-touch gestures for pan, zoom, rotate, tap, double tap, tap and hold (requires multi-touch enabled OS)		
ENHANCED IN 11.3 ALEXANDRIA	TTouchKeyboard a virtual keyboard for enhanced non-keyboard interface interactions that support multiple locales and languages, enhanced with better VCL styling support		
	Modern VCL Components: TCategoryPanelGroup, TButtonedEdit, TLinkLabel, TBalloonHint		
	Full support for PNG image format in TImage and TImageList		
	TDBImage direct support for JPEG and PNG graphic formats		
ENHANCED IN 11 ALEXANDRIA	Support for Windows Imaging Component (WIC) including JPEG, TIFF, GIF, PNG, BMP, and HD Photo image formats. RAW camera formats (such as NRW, NEF, CRW, CR2, RW2, ARW, SR2, SRF, ORF, PEF, and DNG) also supported with manufacturer supplied codecs. Includes multi frame support.		
	Import Component wizard to import a type library, ActiveX control or .NET Assembly		
	Component wizard to create the unit for a custom component		
	Standard components including frame, edit, menu, button, label, checkbox, list box, combo box, panel and action list		
ENHANCED IN 11 ALEXANDRIA	Additional components including graphical buttons, image, shape, scroll box, splitter, buttoned edit, tabs, panels, action toolbar, and color map		
ENHANCED IN 11 ALEXANDRIA	Windows common controls including page control, image list, rich edit, progress bar, date time, calendar, toolbar, tree view and list view		
	System components including timer, paint box, media player, OLE container and DDE		
	Dialog components including open, save, open/save picture, open/save text, font, color, print, printer setup, find, replace and page setup		
	Data Access Controls via datasource and the use of data-aware controls		
	Data Controls including DBGrid, navigator, text, edit, memo, image, listbox, combo box, checkbox, radiogroup, lookup and rich edit		
	TWebBrowser support, wrapped on the Microsoft IE ActiveX component (also offering dual WebView 2 support)		
	Task dialog component and enhanced Windows dialog components for file open and file save		

COM/DCOM SUPPORT

	Full COM/ActiveX support enabling transparent access to IDL and code, source control friendly, and allowing total control over implementation of COM and ActiveX objects; DCOM support		
	Wizards support wrapping existing VCL components as ActiveX controls		
	Import .NET assemblies as COM objects in Win32 VCL applications		

	Legacy support for DataSnap/MIDAS COM/DCOM based multi-tier database application framework		
	Advanced Type Library Editor that manages IDL, making COM development completely transparent; Support for Automation Object event handling		
	Create reusable native 32-bit and 64-bit dynamically linked libraries (.DLL), COM controls (.OCX), and standalone COM executable		
	COM Object Wizard; Microsoft ActiveX® Control Data binding; Import COM servers as components		

AI POWERED FEATURES

NEW IN 13 FLORENCE	SmartCore AI Component Pack - Your Gateway to AI inside Your Apps: a new set of core AI components designed to help our customers add AI features to their apps or build brand new AI-focused apps with Delphi or C++Builder.* ** Available for download in the GetIt Package Manager (not included in Community Edition)*		
NEW IN 13 FLORENCE	The AI-engine agnostic connection component will use specific drivers to communicate via REST with different AI engines (including OpenAI, Claude, Gemini and Ollama). The configuration is managed via an easy-to-use AI connection wizard.		
NEW IN 13 FLORENCE	A set of AI request components for different data types (like text, images, structure data) which can be associated with prompts and can be bound directly with UI controls to display the results.		
ENHANCED IN 13 FLORENCE	Smart CodeInsight offers AI Powered Coding helpers. Smart CodeInsight is an open architecture for AI plugins. Smart CodeInsight UI elements in the IDE include a chat pane (with Markdown support since version 12.3) and an editor menu with direct commands on the selected code. Smart CodeInsight offers an OpenTools API and developers can write plugins for additional LLM engines and APIs. Includes a new Test Connection feature in the configuration dialog.		
ENHANCED IN 12.3 ATHENS	Smart CodeInsight ready-to-use support for three online solutions (OpenAI, Gemini by Google, and Claude by Anthropic) and one offline solution (Ollama). AI plugins can be selectively enabled and configured in a specific Tools Options page, where you can also set the default plugin for the chat and the editor.		
NEW IN 13 FLORENCE	New RAD AI Companion Web Site, a RAD Studio tailored chat bot, offering the ability to query an AI specifically trained on our product and its libraries or to ask it to generate application source code.		

VISUAL LIVEBINDINGS

	Optimized Visual Live Bindings RTTI integration for better performance		
	VCL and FMX controls bound to a TDataSet and TField respect some of the TField properties like Alignment, EditMask, DisplayWidth		
	Visual LiveBindings Designer, to visually connect component properties and data		
	Easily replace prototype data with real ClientDataSet data with one click		
	Context sensitive LiveBindings Wizard supporting both dbExpress and FireDAC		
	Support for QuickBinding components, ability to visually add and bind to additional component properties		
	Color coded component properties easily identify bindable members		
	Zoom control for Visual LiveBindings Designer and ability to save image of current LiveBindings Designer layout; Hide visual elements on right-click		
	Integrated Layers Management system for LiveBindings Designer with Photoshop® like layers, allowing you to visually organize and hide set of elements in the LiveBindings Designer		
	Tools Options support for LiveBindings Designer to show/hide certain components fror the visual LiveBindings Designer view and enable/disable Wizard option in right-click menu		

INTEGRATED BUILD TOOLS AND PROJECT SUPPORT

	IDE support for building and managing projects for 32-bit and 64-bit Windows		

	IDE support for building and managing projects for 64-bit Linux		
ENHANCED IN 11.3 ALEXANDRIA	IDE support for building and managing projects for 64-bit macOS, for Intel and ARM (M1), including universal binary generation, macOS App Store support and integrated Apple Notarization support updated in 11.3 to the process Apple introduced in XCode 14		
	Flexible build system leveraging MSBuild with identical build process from the IDE or command line. Use MSBuild externally from the IDE to compile very large projects with the stand-alone Delphi and C++ compilers.		
	The stand-alone Delphi compilers can use the large memory address space.		
	The IDE shows Header Dependencies when building with bcc32c		
	Define Android application settings, including supported orientations, and use the Entitlement lists to include advertising and notifications support in an app		
	IDE support for creating Android App Bundle (AAB) including both 32-bit and 64-bit binaries for store submission		
ENHANCED IN 11.1 ATHENS	Integrated installation of Eclipse Temurin OpenJDK, required by Android platform tools		
ENHANCED IN 12.2 ATHENS	IDE support for adding Java classes to an Android app classes.dex library and customizing it. In 12.2 we added options to Compress Dex files and Compress native library files, configure targetSdkVersion of the Android manifest file, and added the ability to import Android libraries ('.aar' files).		
	Support for Android Services in the IDE, including wizards to create Android Services and to add them to an existing Android app		
ENHANCED IN 13 FLORENCE	Java2OP (Java to Object Pascal bridge) API headers translation tool (for Android), improved to reduce dependency between units and leveraging updated platform tooling		
	SdkTransform tool to convert Objective-C / C++ headers to Object Pascal headers		
	Define info.plist application settings including supported orientations in the IDE		
	Deploy apps to the iOS Device (Debug/Ad Hoc/App Store), doing iOS device provisioning during the deployment process – including Auto Bundle identifier and support for executing an application from the IDE using the AdHoc target		
INTRODUCED IN 12 ATHENS	Single App Icon Wizard, a comprehensive tool integrated within the RAD Studio IDE, designed to streamline the process of generating icons and splash screens for user applications across the supported platforms		
ENHANCED IN 13 FLORENCE	Deploy apps to Android devices (Debug/Play Store). The process has been improved and optimized in 12.1 and further in 13, with the adoption of newer Android platform tools.		
ENHANCED IN 11 ALEXANDRIA	Platform Assistant for deploying and debugging across platforms, including an updated macOS client (Delphi only)		
	Platform Assistant (PAServer) for Linux; Deployment Manager: Deploy to Linux		
	Platform Assistant Manager macOS tray app that helps managing multiple versions of Platform Assistant (PAServer)		
	Deployment Manager: Deploy to Windows		
	Deployment Manager: Deploy to iOS, Android, macOS		
	Automatic inclusion of project files (like media files and resource files) to the Deployment Manager		
	Project Manager view menu for directory (nested), directory (flat) and list view of files in a project and with an enhanced status bar for more file information.		
	File manipulation and creation through the Project Manager facilities		
ENHANCED IN 12 ATHENS	Delphi 1 through 11 project import and C++Builder 2007 through 11 project import		
	Build Configurations provide flexibility to organize multiple build settings for all compiler, linker options, and platform support		
	Named option sets to save and apply build configurations to any project; Share option-sets between build configurations and between projects		
	Multi-select files in the Project Manager		

ENHANCED IN 11 ALEXANDRIA	Build Events for pre-build and post-build on project and file levels. Enhanced with security warnings.		
	Project dependency checking, Project level build ordering, Build/Make from here – Make or build current project and all later projects		
	Additional command line tools include Microsoft Resource Compiler, TDump utility to structurally analyze EXE, OJB, and LIB files, GREP tool, and MAKE tool		

RECENT C++ LANGUAGE AND CORE RTL FEATURES

ENHANCED IN 13 FLORENCE	Updated Win64 C++ Clang 20 compiler, introducing support for C++23 language specification, and including all of the features previously available for Clang 15		
ENHANCED IN 13 FLORENCE	Updated Win64 C++ Clang 20 STL using libc++, including RTL, C RTL, and toolchain including LLVM-20-based linker. C++Builder 13 also introduces versioned shared libraries.		
INTRODUCED IN 12.2 ATHENS	Win64 C++ Clang parallelised CPU-saturating batch compilation with -jobs		
ENHANCED IN 12.3 ATHENS	Win64 C++ Clang support for consumption and production of runtime packages		
INTRODUCED IN 12.3 ATHENS	Support for Clang "sanitizers", specifically the Address sanitizer and the Undefined Behavior sanitizer.		
ENHANCED IN 11 ALEXANDRIA	Win32 C++ Clang-enhanced compiler (with C++ 17 language support)		
ENHANCED IN 11 ALEXANDRIA	C++ runtime library (RTL) built with the updated Clang-enhanced compiler for Win32 and Win64, including 2018 edition of Dinkumware STL		
INTRODUCED IN 11 ALEXANDRIA	Improved C++ and Delphi code interoperability to assist using Delphi and C++ projects, including string_view support for Delphi strings, easy C++/Delphi string assignment, and string to/from conversion methods for common Delphi RTL types		
ENHANCED IN 11 ALEXANDRIA	Support for using our C++Builder compilers to build CMake projects on the command line for Win32 and Win64, including deployment support		
	Improved optimizations for C++ Clang-enhanced compilers code generation (-O1 and -O2 and also -O3 optimizations)		
	Simplified array initialization and construction for C++ Clang-enhanced compilers		
	The Clang-enhanced compilers are large-address aware and can take advantage of additional memory		
	Additional iterator wrappers over common Delphi RTL containers for all Clang-enhanced compilers for use with C++11 range-for loops and non-member std::begin() and std::end()		
	C++ functions for TPoint, TPointF, TRect, TRectF core types		
	#pragma once support that allows for better compatibility with MSVC and also acts as a header guard		
	[[deprecated]] attribute to flag constructs as deprecated		
	Secure C library functions		
	Custom evaluators for C++ Strings types		
INTRODUCED IN 11 ALEXANDRIA	Using C++ smart pointers (std::make_unique and std::make_shared) on Delphi-style classes		
ENHANCED IN 11 ALEXANDRIA	C++ RTTI compatibility with Delphi, with C++-compatible RTTI generated for Delphi types, including using typeid() on Delphi types		
ENHANCED IN 11 ALEXANDRIA	Updated edition of the Dinkumware STL with C++17 headers for Win32 and Win64		
	ANSI/ISO standard library Technical Report 1		
	Optimized string/memory handling functions, improved standard C++ heap manager		
	UnicodeString class and Unicode character types char16_t and char32_t		
	Move semantics with rvalue references		
	Explicit conversion operators		

	Static assertions	✓	✓
	Full type trait support	✓	✓
	Extern templates	✓	✓
	[[final]] and [[noreturn]] attributes	✓	✓
	Decltype keyword	✓	✓
	Linker error handling to improve memory management flexibility and resolution	✓	✓
	Large-memory space support for C++ linker and improved resource handling to address heap errors	✓	✓

RECENT DELPHI LANGUAGE AND CORE RTL FEATURES

NEW IN 13 FLORENCE	Delphi ternary operator, implemented using the "if" keyword, matching the semantic of the ternary conditional operator found in other programming languages	✓	✓
NEW IN 13 FLORENCE	NameOf intrinsic function, to get the string representation of identifiers in code	✓	✓
NEW IN 13 FLORENCE	New {$PUSHOPT} and {$POPOPT} compiler directives, for taking a snapshot of the current compiler directives and applying them again	✓	✓
NEW IN 13 FLORENCE	New language operators: "is not" and "not in"	✓	✓
NEW IN 13 FLORENCE	New "noreturn" directive for procedure declaration	✓	✓
INTRODUCED IN 12 ATHENS	Long string literals and multiline string literals, limited by triple quotes, to allow embedding of SQL, HTML, JSON, XML and similar text within an application source code	✓	✓
INTRODUCED IN 12 ATHENS	Weak type alias definition for NativeInt, do help write code that works better across 32-bit and 64-bit compilers.	✓	✓
INTRODUCED IN 12 ATHENS	Support for generating a GraphViz file with the uses unit graph, to help detect circular uses reference and visualize the overall structure of the dependencies across the modules of an application	✓	✓
INTRODUCED IN 12 ATHENS	Support for NaN comparisons for floating point numbers, as required by IEEE 754	✓	✓
INTRODUCED IN 11 ALEXANDRIA	Binary literals and digit separator support. Inline assembler support for AVX instructions (AVX-512)	✓	✓
ENHANCED IN 13 FLORENCE	Custom managed records with the ability to define custom Initialize, Finalize and Assign operations, executed when a record is allocated, de-allocated, and copied. Improved in 13 with implicit "Self" parameter declaration in record class operators Initialize and Finalize.	✓	✓
	Standard functions for low-level bit processing operations: CountLeadingZeros, CountTrailingZeros, CountPopulation	✓	✓
	Inline variables declaration, supporting local scope, direct initialization, and type inference; inline constants declaration	✓	✓
	Type inference for inline variable declarations (the type is inferred from the value assigned to the variable)	✓	✓
ENHANCED IN 13 FLORENCE	Support for calling WinRT APIs, with over 50 specific Object Pascal interface units. The set of headers has been updated in version 13	✓	✓
	[Weak] and [Unsafe] support for interface references	✓	✓
	[Volatile] attribute can be attached to a method parameters and local variables	✓	✓
	Improved initialization for dynamic arrays, support for mixing and matching constant and dynamic arrays	✓	✓
	String-like operations on dynamic arrays including + operator concatenation, Insert and Delete	✓	✓
	Class, record, and intrinsic types helpers	✓	✓
ENHANCED IN 11 ALEXANDRIA	Predefined helpers for core types such as string, char, integer, other ordinal types, and floating point types. Available also for TDateTime and Currency data types.	✓	✓
ENHANCED IN 11 ALEXANDRIA	RTTI for indexed properties and RTL support, plus invoking methods using open arrays	✓	✓

	Reflection support for exposure of Methods, Fields, and Properties to support dynamic invocations and other meta-programming approaches (Enhanced RTTI)	✓	✓
	Custom attribute support for most code elements – types, fields, properties, methods, and parameters	✓	✓
	Generics with full RTL list and collection support	✓	✓
	UnicodeString type as the default string type and support for Unicode char manipulation	✓	✓
	Anonymous methods (or closures)	✓	✓
	Compiler option to treat warnings as errors	✓	✓

SHARED RUNTIME LIBRARY FEATURES

	Unified memory management across all Delphi compilers and platforms, using classic Delphi memory management also for mobile (and Linux, since RAD Studio 10.3)	✓	✓
	Delphi RTL for 64-bit Linux		✓
ENHANCED IN 11 ALEXANDRIA	Delphi and C++ RTL for 32-bit Windows and 64-bit Windows	✓ ✓	✓ ✓
	Delphi and C++ packages support for 32-bit applications and 64-bit Windows	✓ ✓	✓ ✓
	Support for Windows 10 Notifications using the NotificationCenter component	✓ ✓	✓ ✓
	Native Bluetooth APIs (on platforms that fully support the technology) for FireMonkey and VCL applications	✓ ✓	✓ ✓
	Bluetooth framework improvements and TBluetooth component for classic Bluetooth	✓ ✓	✓ ✓
ENHANCED IN 12.3 ATHENS	Bluetooth LE component (on platforms that fully support the technology)	✓ ✓	✓ ✓
	Proximity support based on the "beacons" technology (including the iBeacon and AltBeacon standards) for all supported platforms	✓ ✓	✓ ✓
	Beacon enhancements with support for EddyStone format, change of AdvertiseData parser to include all fields, improved Beacon scan performance, Android scan modes, and WinRT Bluetooth API integration, StartScan method on Android. Android support is Delphi only	✓ ✓	✓ ✓
	TBeaconDevice class for turning a device on one of the supported platforms into a "beacon"	✓ ✓	✓ ✓
	Support for Windows Store interaction via TWindowsStore component, with support for paid applications, paid add-ons, and trial mode	✓ ✓	✓ ✓
	Support for contracts, the system mechanism for sharing information with other Windows 10 applications (apps can behave like *contract sources*) using the SharingContract component	✓ ✓	✓ ✓
	Parallel Programming library for easy development of asynchronous code and faster multi-core applications, without coding threads manually	✓ ✓	✓ ✓
	System.Threading unit supporting parallel for loops, tasks scheduling and futures	✓ ✓	✓ ✓
	Optimized Unicode strings handling in iOS and Android ARM compilers	✓	✓
	Support for all classic string types (including UTF8String, ANSIString, and RawByteString) on iOS and Android	✓	✓
ENHANCED IN 11 ALEXANDRIA	Delphi RTL for macOS 64-bit (Intel and ARM M1)	✓	✓
	Delphi RTL for iOS, iOS 64-bit API headers and RTL integration	✓	✓
ENHANCED IN 12.3 ATHENS	Delphi RTL for Android, both 32-bit and 64-bit. With support for Android API level 35, a coming Google requirement for the Play Store app submission.	✓	✓
INTRODUCED IN 12.3 ATHENS	Support for 16 KB page sizes on Android (a requirement for Android 15).	✓	✓
	DirectX 12 support on Windows	✓ ✓	✓ ✓
	OpenGL 4.3 support	✓ ✓	✓ ✓
	Enhanced compiler switches for XML representation of source code	✓ ✓	✓ ✓

	Extensible, cross platform, framework agnostic login credentials service	●	●
ENHANCED IN 11 ALEXANDRIA	Native zip file support. TZipFile class has a callback to use during the extraction process to know the current progress, support added to extract ZIP files with passwords. With Zip64 support and additional features.	●	●
ENHANCED IN 11.2 ALEXANDRIA	Regular Expressions (Regex) library, supporting PCRE UTF-16 on Windows. PCRE is compiled with JIT enabled for faster regular expressions execution.	●	●
	Object-oriented path, file and directory I/O classes	●	●
	Buffered file read and write support, based on the TBufferedFileStream class, a high performance TFileStream descendant based on an internal configurable buffer	●	●
	TMemIniFile and TIniFile have Modified and AutoSave properties	●	●
	TStrings enhancements including properties IsUpdating, TrailingLineBreak, UseLocale, Options, plus overloaded TStringList constructors and AddPair fluent method	●	●
	StringReplace and Pos speed optimization	●	●
	TStringBuilder for easier and faster string concatenation, optimized and with flexible growth strategy	●	●
	JSON (JavaScript Object Notation) processing using a JSON.NET implementation for JSON streaming with readers and writers (including base TJsonReader and TJsonWriter classes)	●	●
	Specialized JSON readers and writers, including the TJsonTextReader and TJsonTextWriter classes, the TAsciiStreamWriter class, support for Extended JSON, better exception handling when reading not-well-defined JSON	●	●
	Binary JSON (BSON) readers and writers support, as part of the same JSON.NET architecture (including the TBsonReader and TBsonWriter classes)	●	●
	JSON and BSON fluent method builders, including the TJSONArrayBuilder and TJSONObjectBuilder classes	●	●
	JSON and BSON fast forward-only iterator (TJSONIterator)	●	●
INTRODUCED IN 12 ATHENS	JSON mapping wizard, which can generate classes matching a JSON data structure and proper reading and writing code to read the JSON in a set of objects and stream it out to a JSON file	●	●
ENHANCED IN 11 ALEXANDRIA	System.NetEncoding RTL unit for Web encoding and decoding (Base64, HTML, URL, Base64URL)	●	●
	System.Hash RTL unit, with hash functions to support the HTTP framework, including also method for files hashing	●	●
	HTTP client framework, mapped to platforms libraries on all supported platforms, including also platform HTTPS support (with no need to deploy the OpenSLL library), with improved asynchronous support	●	●
ENHANCED IN 11 ALEXANDRIA	NetHTTPClient and NetHTTPRequest components, offering easy access to the HTTP client framework, with HTTP / 2 protocol version support	●	●
	Support for native HTTP(S) client libraries on Windows	●	●
	Support for native HTTP(S) client libraries on macOS, iOS and Android	●	●
	Support for native HTTP(S) client libraries on Linux		●
	HTTP client library improvements including support for asynchronous HTTP requests, time-outs, and automatic decompression of gzip content	●	●
	Support for localized resources	●	●
	Expression evaluation in compiler directives	●	●
	Custom variants with support of your own data types, complex numbers, safe arrays, and passing variants through data sets; Variant support for Int64, unsigned types and Unicode strings	●	●

INTEGRATED DEVELOPMENT ENVIRONMENT

ENHANCED IN 13 FLORENCE	RAD Studio 64-bit IDE, offering support for the Windows 64-bit targets platform. The 64-bit IDE is installed side by side with the 32-bit IDE, which is still the default version. Now including: code tooling for C++ via Visual Assist; support for COM, ActiveX and	●	●

	Type libraries; Live Templates (Ctrl+J) and the Surround menu, and quality improvements.		
ENHANCED IN 12.2 ATHENS	Editor split views: multiple editors side by side, above/below each other; split editors allow same file to be edited next to itself, keep both a design and a code view on screen		
INTRODUCED IN 12 ATHENS	Improved UX for the feature installer, used for the initial product installation or to add additional features and platforms from the IDE. Includes extended logging and easier to access log files.		
INTRODUCED IN 12.2 ATHENS	RAD Studio Focus Mode (aka 'zen mode') leaves the IDE with the editor(s) only, with reduced, essential-only UI, so you can avoid interruptions to your flow of thoughts, and do not become distracted by the many IDE panes and features		
INTRODUCED IN 12.2 ATHENS	With scrollbar annotations, the editor scrollbar shows at a glance the sections of the code which have been modified, have bookmarks, have errors or warnings, and much more. This is powered by an OpenTools API and can be enhanced by developers.		
NEW IN 13 FLORENCE	Extensive and configurable search features in many IDE panes, including the Project Manager, the Structure View, the Message pane and Events log		
NEW IN 13 FLORENCE	Additional highlighting options in the Message Log and grouping for compiler hints and warnings in the same source code file.		
ENHANCED IN 11.3 ALEXANDRIA	The RAD Studio IDE is high-DPI enabled (using perMonitor v2 Windows configuration). All primary forms and panes have been enhanced, with high-DPI enabled icons and toolbar button images, smooth IDE and editor fonts, and a full high-DPI experience. It remains possible to run the IDE in low-DPI, classic mode.		
ENHANCED IN 11.3 ALEXANDRIA	Both the VCL and the FireMonkey designers have been enhanced to fully support designing applications in high-DPI. The VCL designer offers multiple configurations and options for improved compatibility.		
ENHANCED IN 11.2 ALEXANDRIA	VCL styles preview in the designer offers the ability to design VCL forms using styles and enabling per-control and per-form styling. This feature is enabled by default.		
ENHANCED IN 11 ALEXANDRIA	Code and design a form at the same time in multiple windows, including moving the designer between windows		
INTRODUCED IN 11.3 ALEXANDRIA	The IDE title bar shows when the it is running elevated, as administrator		
ENHANCED IN 11.2 ALEXANDRIA	Welcome Page with rapid access to commonly used features, including recent and favorite projects, GetIt and YouTube feeds, patches notifications. The layout of the Welcome Page is highly configurable, offers background images, and supports adding custom frames with a specific ToolsAPI.		
INTRODUCED IN 11.2 ALEXANDRIA	IDE support for Markdown documents, including Markdown preview, and a VCL-based HTML preview (replacing the Internet Explorer based one)		
INTRODUCED IN 11 ALEXANDRIA	FireMonkey designers guidelines and designer hints (component hints and position hints)		
ENHANCED IN 12.2 ATHENS	Delphi Code Insight implementation (based on the Language Server Protocol architecture, or Delphi LSP) for including Code Completion, Parameter Completion, Error Insight, Find Declaration, and Tooltip/Help Insight. Enhancements include support for include files, better lifetime management of the agent processes, numerous fixes, and increased performance. In 12.x, code completion again includes language keywords and code templates, the ability to customize the behavior of specific acceptance keys, while 12.2 has improvements in auto completion and navigating symbols in the class interface and forwarded type declarations.		
INTRODUCED IN 12.2 ATHENS	64-bit binary version of the Delphi LSP engine, to allow parsing a very large project with no memory constraints		
REINTRODUCED IN 13 FLORENCE	Delphi Classic CodeInsight for code completion, available again.		
NEW IN 13 FLORENCE	New Delphi Upgrade Advisor wizard, which allows developers to identify meaningful changes to project configuration and code that can significantly improve the compilation time and code tooling experience.* ** Available for download in the GetIt Package Manager (not included in Community Edition)*		
ENHANCED IN 11.3 ALEXANDRIA	Help Insight implementation, based on customizable XSLT and CSS configurations and displayed in a VCL-based HTML view		
ENHANCED IN 12.1 ATHENS	C++Builder integration of the Visual Assist, a plugin previously available only for Visual Studio, offering better Code Insight including code completion, code navigations and refactorings.		

ENHANCED IN 13 FLORENCE	Visual Assist features include finding where your code is used with Find References or Find Symbol, refactorings which assist maintenance and active development in code, and navigation including the Go To Related feature. See the Refactoring section for more information. Available also in 64-bit IDE.		
INTRODUCED IN 11 ALEXANDRIA	C++ code formatter based on clang-format, including the ability to format as you type and several predefined formatting styles		
INTRODUCED IN 11 ALEXANDRIA	Support for using Delphi LSP in Visual Studio Code and other external editors		
ENHANCED IN 11.2 ALEXANDRIA	Enhanced editor tabs, including the ability to use different, customizable colors for the tabs of files of different types and more options for the close button.		
INTRODUCED IN 11.3 ALEXANDRIA	Editor ability to highlight matching words. This feature allows you to see instances of the same word you have selected, or the cursor is currently in, highlighted on the screen.		
	Code editor status bar font zooming, syntax highlighter and file encoding selections		
INTRODUCED IN 12 ATHENS	Syntax highlighting in many secondary windows, like Structure view, Error Insight messages, and Navigation toolbar.		
	Modern IDE Title bar with integrated IDE desktop functionality (using the matching VCL control)		
	IDE productivity tool: Bookmarks*, which extends the IDE's previous marking of locations in the code editor. ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	IDE productivity tool: Navigator*, which allows you to quickly jump to any location in your unit through intelligent search (for Delphi), and adds a minimap to the code editor. ** Available for download in the GetIt Package Manager (not for Community Edition)*		
ENHANCED IN 11.2 ALEXANDRIA	Styled IDE UI, including ability to toggle and disable IDE styling and support for a dark style. The IDE has a Light theme based around the blue color, a Dark theme and a third Mountain Mist theme. Since 11.2 it's possible to also use a custom VCL style.		
	Additional IDE Startup Layout for the Welcome Page, with the ability for users to change and set the layout they want to use for each layout, including Startup		
INTRODUCED IN 11.1 ALEXANDRIA	The Messages view uses different colors for compiler errors, messages, and hints to better highlights issues (the colors can be customized)		
	Delphi project option "Show general messages" to show info messages when building a project		
	IDE built with large memory address model, to provide significantly more memory to the embedded compilers, integrated debuggers, and various tools executed in the IDE process		
	Compiler unit caching mechanism, to free older units in case the available memory is being filled up, when compiling project groups with multiple projects		
	Multi-monitor support in the IDE, with the ability to place most forms and panes on different secondary monitors		
	Form designer option to hide/show non-visual controls icon (reducing form design potential clutter)		
	Structure View Icons representing the corresponding component		
	Unsaved file auto-recovery – unsaved work is periodically saved to a temporary location. If the IDE closes ungracefully (because of a system crash) the IDE will attempt to recover the user's unsaved work		
	IDE Project Options to enable High-DPI Awareness in your applications, supporting several execution levels affecting the project manifest file on Windows		
	Full customization of the Object Inspector layout, with the ability of hiding the description panel, the quick actions, and the filter panel		
	Object Inspector contents can be filtered to display specific elements		
	Shortcuts to increase/decrease the size of the font in the code editor		
ENHANCED IN 13 FLORENCE	GetIt Library Manager for easy discovery, download and update of source code libraries, components, and other features from Embarcadero GetIt servers*. Enhanced in 13 with package versioning support.		

	* Use of GetIt requires Internet connectivity. The feature is not available to off-line developers. A small selection of packages is available as a separate download and can be installed using the GetIt Local Packages support.		
ENHANCED IN 13 FLORENCE	GetIt Library Manager with support for versioning, updates, sort by date, subscription only packages, cached images, and an updated UI. Enhancements in 11.2 included a filter by language and the introduction of Local GetIt Packages*. Further enhancements in 12 include the ability to load multiple GetIt packages* at once and better logging. ** The us of Local GetIt Packages requires an active Update Subscription*		
	Configuration settings migration tool to migrate configurations of older versions of the product to a newer version, or between different installations of the same version		
	Project statistics information for activity tracking and to better understand team productivity (optional and disabled by default)		
	Multi-paste support to let you perform the same paste operation on multiple source code lines at once		
ENHANCED IN 12 ATHENS	Structural highlighting ; Flow control highlighting, to visually see the flow jumps in your code directly in the editor; Code navigation toolbar with classes and methods combo boxes at the top of the editor, with an additional option to hide it		
	Support for adding database files and simplifying their deployment in the Project Manager		
ENHANCED IN 11 ALEXANDRIA	Install experience based on the GetIt architecture; with the ability to select the platforms to install and add more platforms later from the IDE (which has a Platforms Manager option in the Tools menu). Recent features include upfront language selection and custom catalog repository folder, plus a faster parallel download and installation Supporting unattended, silent installation both in online and offline mode.		
	A set of examples and demos to help new and existing developers get the most out of the environment		
	Designer snapshots – copy form images to the clipboard		
	IDE Insight, the fastest way to find and execute commands in the IDE with 1-button click also with object inspector property support		
ENHANCED IN 11 ALEXANDRIA	Command-line Code Formatter for integration with automated build processes, enhanced in 11 Alexandria with clang-format support for C++	PROFESSIONAL only	
INTRODUCED IN 11 ALEXANDRIA	.clang-format C++ code formatter configuration file support to automatically format code following the per-project convention used by many open source projects, plus a Embarcadero style		
	Search for Usages command available from the code editor and Delphi Class Explorer context menus		
	Compilation can be done in a background thread		
	Support for C++ parallel building through batch compilation		
	Build a C++ project accelerated over multi-core with the TwineCompile addon ** Available for download in the GetIt Package Manager (not for Community Edition)*		
	Checkbox for Boolean types in the Object Inspector greatly enhancing readability		
	Class Explorer gives configurable hierarchical views of class libraries throughout the project. Also enables fast navigation to declarations and implementations, as well as easy addition of fields and methods.		
	Resource Manager to simplify the addition of resources to projects		
	Live Code Templates – dynamic scriptable templates streamlines coding complex and common structures and operations. Create or customize code templates and control the behavior, code generation, and user experience of your templates. Surround Templates that can wrap around selected code or text.		
	Editor "Change Bar" indicates modified lines and saved modified lines		
	"Smart" Block completion – anticipates and gracefully adds block closures as you type		
	Virtual form positioner for VCL designer and FireMonkey designer		
	Main toolbar with the addition of Run without Debug option		

	Ability to run selected applications with either Run or Run without Debug from the context menu in the Project Manager	✓	✓
	Hot-key Tool Palette w/ Incremental search – find any component quickly	✓	✓
	Gallery in Tool Palette – find gallery wizards when you need them	✓	✓
	SyncEdit – block-edit multiple instances of a symbol simultaneously	✓	✓
	History Manager with automatic multi-level file backups, file differencing and restore	✓	✓
	Structure Pane that displays hierarchical view of source code and provides a VCL Object Treeview for quick navigation and manipulation of objects in the VCL designer	✓	✓
	Syntax Highlighting Open Tools API extension with built in support for JavaScript, PHP, and INI files	✓	✓
	Ability to individually customize editor options for particular file types	✓	✓
	Full-featured Open Tools API allows for customization of the IDE through wizards and experts, operate on projects and their various modules, access to the editor and designers, with improved style and theme support	✓	✓
ENHANCED IN 13 FLORENCE	IDE ToolsAPI further extended in 13. In 12 it added direct access to editor content and more information about the editor display. Version 11.3 added support for painting in the code editor and highlighting elements of the source code, a foundation allowing developers to customize the editor.	✓	✓
	Symbol Insight – click to navigate to declarations and implementations	✓	✓
	Code Folding configuration and properties	✓	✓
	Updated support for Code Completion for the C++ language, with asynchronous invocation support, based on the Language Server Protocol architecture		
	Custom IDE desktop layout profiles with enhanced support for docked windows	✓	✓
	To-do lists keep your development on track	✓	✓
	Help Insight – convenient tooltips provide help snippets as you develop	✓	✓
	English, German, French, and Japanese translations available for IDE menus and dialogs, compiled units, resources, and source code	✓	✓
	Easy switching between the available languages for the IDE menus, dialogs, and more	✓	✓

SOURCE CODE MANAGEMENT

	Version Insight – Integrated framework for supporting source code management tooling inside the IDE	✓	✓
	Subversion integration into the IDE, including change list labels, SVN merge dialog, branches switching and more (requires a separately installed subversion client library)	✓	✓
	Subversion/History integration	✓	✓
	A full default Diff viewer	✓	✓
	Git version control integration into the IDE, including authentication, Push and Pull changes to and from remote repositories	✓	✓
	Mercurial version control integration into the IDE (supports cloning your remote repository, commit changes locally, show log data)	✓	✓
	Version Insight also available as an open source project	✓	✓

REFACTORINGS

ENHANCED IN 12.1 ATHENS	Rename Refactoring across code and designers for C++, provided by the integration of Visual Assist		
INTRODUCED IN 12.2 ATHENS	Three refactorings powered by Visual Assist: Introduce Variable, Create From Usage and Extract Method.		
INTRODUCED IN 12 ATHENS	Find References, provided by Visual Assist. Configurable Find References granularity: entire project group, current project, or the current unit		

INTRODUCED IN 12 ATHENS	Create Declaration and Create Implementation, to automatically create the matching method declaration or empty method implementation when only one is defined. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Add Include: adds the #include statement to include the header for a type used in code, for both user headers and system headers. Provided by the integration of Visual Assist		

CODE SEARCH AND NAVIGATION

	Find and Replace inline in the code editor, including incremental search		
	Find in Files, with the ability to search using a search term, including optional case sensitivity or for whole or partial words, or a regular expression across the current file, files in the project, project group, open files, or folders		
	Find Declaration: go to the declaration of a type or method		
INTRODUCED FOR C++ IN 12.2 ATHENS	Go To Interface and Go To Declaration: navigate to the declaration or implementation of a method, including quick toggle shortcut Ctrl+Shift+Up/Down. For C++, provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Go To Member, to search and navigate through the member fields, methods and properties of a type, including of the type of a variable. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Find Symbol, to find any symbol defined in the project group and any headers including system and platform headers it includes. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Go To Related: for #include files, to list and navigate to the headers included by the header in question, or the headers that include the header in question, as well as opening the file or showing it in Explorer. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Go To Related: for types, list and navigate to the ancestor base classes, or derived descendant classes; go to the constructor, type definition, or type declaration; or go to member. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Go To Related: for methods, go to the base / ancestor methods a virtual method overrides, or the descendant derived methods that override a virtual method; go to type member. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Go To Related: for variables / parameters: go to the declaration of the symbol's type, including portions of the type such as template and template parameter; go to type member. Provided by the integration of Visual Assist		
INTRODUCED IN 12 ATHENS	Go To Related: similar features for macros and properties defined using the __property keyword, including going to ancestor and descendant property redeclarations. Provided by the integration of Visual Assist		

INTEGRATED DEBUGGING

ENHANCED IN 12 ATHENS	Full-featured debugger with color syntax highlighting for the call stack, debugger tooltips and other views		
	Debug remote and local 64-bit Windows applications		
ENHANCED IN 13 FLORENCE	Debugger for C++ for Win64, based on LLDB, including formatters to evaluate common complex types like UnicodeString and AnsiString, C++ strings, vectors, deques, and more. Updated to LLDB 20.		
ENHANCED IN 13 FLORENCE	Delphi Win64 LLDB-based debugger in the 64-bit IDE. Updated to LLDB 20.		
ENHANCED IN 11.3 ALEXANDRIA	Debug remote macOS 64-bit applications (Intel and ARM). The Delphi macOS 64-bit ARM debugger is based on the LLDB debugger architecture, and in 11.3 the Delphi macOS 64-bit Intel debugger uses LLDB as well.		
ENHANCED IN 11.2 ALEXANDRIA	Remote iOS 64-bit device debugging (excluding iOS 17 devices, where debugging is currently unavailable)		
INTRODUCED IN 11.2 ALEXANDRIA	Debugging of apps on iOS Simulator on ARM 64-bit macOS (based on LLDB debugger architecture)		
ENHANCED IN 11.1 ALEXANDRIA	Debug Android 64-bit apps on device. The Delphi Android 64-bit debugger is based on the LLDB debugger architecture,		

ENHANCED IN 11.2 ALEXANDRIA	Remote Linux 64-bit debugging. The Delphi Linux 64-bit debugger is based on the LLDB debugger architecture,		
	Improved debugging support for Clang-enhanced C++ compilers for Win32		
NEW IN 13 FLORENCE	Attach to Process dialog box redesigned to include additional columns that display information about processes		
ENHANCED IN 12.2 ATHENS	Debugging support for the Clang 15 Win64 C++ compiler, with support for PDB debug info		
	Support for Delphi Unicode string evaluation, for evaluating properties and variants and function call support when debugging iOS64 applications		
	CPU View support for iOS and Android application debugging along with support for Delphi Dynamic Arrays		
	Debugging visualizers to enable extended views of debugging data, enhanced with support for Delphi generics and C++ templates		
	Build debugger visualizers for your own custom data types		
	Debug visualizer for TStrings, TDateTime, TDate, and TTime types		
	Detach from “Debug session in progress. Terminate?” dialog		
	Multi-threaded-debugging the ability to selectively “Freeze” and “Thaw” threads		
	Multi-threaded-debugging the ability to set a breakpoint for a selected thread		
	Support for Wait Chain Traversal on Windows		
	Disassembly pane has “Show Opcodes” and “Show Addresses” local menu items		
	Debugger Options: “Scroll new events into view” and “Ignore non-user breakpoints”		
	Call Stack View shows a glyph indicating if the frame has debug info		
	CPU view panes can be opened individually outside of an editor tab		
	Integration between debugger views and panes making it easier to find and understand information		
	Call Stack view automatically syncs the Locals view when you double click an item		
	Evaluator tooltip hints go transparent when the CTRL key is pressed, allowing you to see through them to the editor		
	Expandable tooltip expression evaluation		
	Expandable watches and local variables view		
	Selective symbol table loading		
	Close implicitly opened files after debugging		
	Multi-select and copy to clipboard support in CPU view		
	Ctrl-Click in editor gutter to enable/disable breakpoints		
	Sort by load order in Modules view		
	Set Next Statement local menu item in editor		
	Exception notification dialog: Break/Continue buttons, “Ignore exception type” checkbox		
	Thread view displays status on all processes and threads under debugger control. Multi-Process debugging for easy tracing through complex applications. Debug spawned processes		
ENHANCED IN 11.2 ALEXANDRIA	CPU view for low-level debugging, Starting with 11.2, the CPU View offers syntax highlighting		
	Breakpoint view with toolbar, in-place editing, and enable/disable checkbox. Breakpoint Data Aware Triggers. Advanced breakpoints with Tooltips, actions, and groups for complete debugging control		

	Watch view supporting in place editing, checkbox enable/disabling, and the watch names separated from the values with columns	✓	✓
	Multi-tab watch view for logically grouping variables	✓	✓
	Attach to and debug any process, detach from any process	✓	✓
	Run until return	✓	✓
	Debug Inspector for easily monitoring properties while debugging	✓	✓
	Event log view	✓	✓

INTEGRATED UNIT TESTING

	Integrated unit testing with the DUnitX testing framework	✓	✓
	DUnitX unit testing support for mobile platforms (iOS and Android)	✓	✓
	DUnitX unit testing support for Linux		✓
	Runtime support for the DUnit testing framework	✓	✓
	Unit test wizards for quick and easy unit test and unit test project creation	✓	✓
	Execute unit tests from project manager	✓	✓

INTEGRATED HELP

NEW IN 13 FLORENCE	New RAD AI Companion Web Site, a RAD Studio tailored chat bot, offering the ability to query an AI specifically trained on our product and its libraries or to ask it to generate application source code.	✓	✓
	Support for CHM help engine and updated IDE help structure	✓	✓
ENHANCED IN 12 ATHENS	Continually updated help system with diagrams showing class hierarchy and declared members pages	✓	✓
ENHANCED IN 12 ATHENS	Online documentation available at https://docwiki.embarcadero.com/	✓	✓

DATABASE APPLICATION DEVELOPMENT

	Powerful cross-platform and multi-database DB access architecture, based on TDataSet class, field definitions, fields and parameters managements	✓	✓
	Support for multiple TDataSet descendant classes shipping with the product or available form third-party vendors	✓	✓
	Support for display options for BlobFields	✓	✓
	Direct GUID access for TField classes	✓	✓
	Control of the automatic and persistent fields of a data set using the FieldOptions and FieldLifeCycle features.	✓	✓
NEW IN 13 FLORENCE	Expanded support for Unsigned Big Integer database fields	✓	✓

FIREDAC MULTI-DEVICE DATA ACCESS LIBRARY [2]

	Desktop platforms enterprise database support, including MySQL, MariaDB, Microsoft SQL Server, Oracle Database, InterBase, PostgreSQL, Informix, Sybase SQL Anywhere, Microsoft Access, IBM DB2 Server, Firebird, Advantage Database, generic ODBC driver		✓
	Support for local databases, including Microsoft Access database, SQLite and MariaDB database, IBToGo / IBLite, InterBase on localhost, MySQL Embedded, MySQL Server on localhost, Advantage Database local engine, PostgreSQL on localhost, Firebird Embedded, Firebird on localhost *Some features in this table marked as [2] are limited to local connections in Professional and Community Editions*	✓	✓

[2] *Some features are limited to local connections in Professional and Community editions*

	Linux enterprise database support, including MySQL, MariaDB, Microsoft SQL Server, Oracle Database, InterBase, PostgreSQL, Sybase SQL Anywhere, IBM DB2 Server, Firebird, Advantage Database, MongoDB, generic ODBC driver. Includes also support for local databases like SQLite database and IBToGo / IBLite.		✓
INTRODUCED IN 12 ATHENS	FireDAC support for the classic QBE mode (Query-By-Example), making it easier to let customers create database table filters	✓	✓
INTRODUCED IN 12 ATHENS	FireDAC configuration (TFDSecurityOptions) to help developers improve database application security with options to filter allowed SQL command kinds, disable single SQL statements with multiple commands, disable dynamic changes to SQL statements, and verifying operations affect only one row.	✓	✓
ENHANCED IN 11.2 ALEXANDRIA	Improved FireDAC's FDMonitor and FDExplorer tools	✓	✓
ENHANCED IN 12 ATHENS	iOS and Android mobile database support, including SQLite, IBToGo and IBLite. IBLite is not available also in the iOS Simulator	✓	✓
ENHANCED IN 12.3 ATHENS	FireDAC Microsoft SQL Server driver, with support for the latest Microsoft ODBC Driver for SQL Server (version 18) and Microsoft SQL Server 2022.		✓
ENHANCED IN 12.3 ATHENS	FireDAC MySQL driver, supporting MySQL v 8.4 and MariaDB v 10.6, MySQL Prepared Statements API, internal BLOB streaming	✓[2]	✓[2]
	FireDAC ODBC driver 13, with support for data type mapping by data type name		✓
ENHANCED IN 12 ATHENS	Updated FireDAC SQLite driver with support for version 3.42 while maintaining the option of using SQLite 3.31.1 with FireDAC encryption (FDE). Offers both static and dynamic linkage of SQLite engine. Includes support for SQLite SEE.	✓	✓
	FireDAC InterBase with support for TRUNCATE command, transaction wait time and little/big endian configuration in connection parameters, which adds to the support for querying the database service manager, real BLOB streaming and array data type	✓[2]	✓[2]
ENHANCED IN 13 FLORENCE	FireDAC Oracle driver to support v 19c and 128-character parameter names for Oracle Stored Procedures. Previous updates included, among many features, support for VARCHAR2, NVARCHAR2, and RAW data types up to 32K, native BOOLEAN in PL/SQL, 64-bit integers and unsigned INTs, identity / auto-increment columns, query change notifications, implicit ROWID fetching, implicit results, additional connection modes, network timeouts, and transaction state detection and synchronization. Since version 12.3, with support for Oracle DRCP (Database Resident Connection Pooling). Version 13 adds OAuth authentication for Oracle.		✓
	TFDOracleAdmin service component		✓
	FireDAC Advantage driver with support for version 12 features like GUID data type, backup archive, default values phrase; improved support for encrypted tables, improved TFDADSUtility component	✓	✓
	FireDAC DB2 driver, with support for DB2 on AS/400, IDENTITY columns, and a connection definition parameter		✓
ENHANCED IN 13 FLORENCE	FireDAC Firebird 5 driver. Includes with VendorHome driver configuration, support for Firebird embedded and local connection protocol, FB$OUT package, long statements, little/big endian configuration, Firebird 4 data types, FDFBBackup parallel operations for Firebird 5. Enhanced with parallel backup operations support.	✓[2]	✓[2]
ENHANCED IN 13 FLORENCE	FireDAC PostgreSQL driver for PostgreSQL up to version 16. Past updates included PostgreSQL 15 specific performance improvements. support for PostgreSQL Stored Procedures, identity columns, macaddr8, password encryption and little/big endian configuration in connection parameters. Enhanced with support for array types.	✓[2]	✓[2]
	FireDAC SQL Anywhere driver, with support for ToolHome property and FDEventAlerter		✓
	Updated FireDAC Informix driver with support for retaining update locks and pessimistic locking, plus ISAM error codes		✓
ENHANCED IN 12.3 ATHENS	Complete FireDAC support for the NoSQL MongoDB database, including a FireDAC MongoDB driver, with support for versions 3.0 to 8.0, including MongoDB Atlas		✓
	MongoDB API wrapping classes, including TMongoConnection, TMongoDatabase and TMongoCollection		✓
	MongoDB query, pipeline, update commands, with fluent methods builders		✓
	MongoDB specific datasets, including TFDMongoDataSet (which allows to attach dataset to MongoDB cursor), TFDMongoQuery (which allows to execute queries to MongoDB collection) and TFDMongoPipeline (which allows to execute pipelines to		✓

	MongoDB collection), and TMongoDocument Iterator method to iterate through document items		
	FireDAC Teradata database support, based on the Teradata database ODBC driver		✓
	FireDAC support for InterBase Change Views	✓[2]	✓[2]
	FireDAC support for change notifications, with additional support for Oracle and MongoDB databases		✓
	Tools and scripts to help migration of dbExpress code to FireDAC	✓	✓
	FireDAC streaming support for blob fields and support for MSSQL file streams		✓
	Support for DBMS API command native timeouts	✓	✓
	FireDAC connection parameters are displayed in the Object Inspector as a record	✓	✓
	IBLite specific FireDAC driver for desktop and mobile (Delphi only)	✓	✓
	InterBase encryption connection options	✓	✓
	Windows and macOS (Delphi only) native driver for Informix database		✓
	Source code of the included database drivers		✓
	Comprehensive support for SQLite v3.+ database, including "smart" data type recognition, supporting both dynamic and static linking	✓	✓
	FireDAC ETL support (Batch move) with optimization and extension of the TFDBatchMoveSQLWriter component, with support for MERGE / INSERT OR REPLACE commands and for dmAppend, dmUpdate, dmAppendUpdate modes. Features the JSON Writer support added in 10.2.2.	✓	✓
	Easy to use TDataSet descendant classes	✓	✓
	Unified Data Access API	✓	✓
	FireDAC cached updates mode, offering AutoCommitUpdates property and compatibility with ClientDataSet	✓	✓
	FireDAC clients for DataSnap		✓
	High-performance in-memory dataset	✓	✓
	FireDAC complete source code, including drivers source code		✓
	Live Data Window mode enabling fast bi-directional navigation through large datasets	✓	✓
	Array Data Manipulation Language (DML) command execution and Command Batches for batch applications and for minimizing network traffic	✓	✓
	Direct support for asynchronous command execution, command execution timeout, and command execution canceling	✓	✓
	Cached updates mode with ability to track correlated changes for several datasets with cascading updates	✓	✓
	Full support for auto-incrementing fields, including those based on generators and table triggers	✓	✓
	SQL dialect abstraction through FireDAC escape sequences, conditional statements and macros	✓	✓
	Data type unification with flexible and adjustable data type mapping	✓	✓
	Easy to use TFDMemTable - in-memory dataset, highly compatible with TClientDataSet, including optimized JSON serialization and the ability to edit the dataset data at design time	✓	✓
	DataSnap client support for HTTP, HTTPS and TCP/IP protocols and authentication		✓
	Local SQL with heterogeneous SQL commands to TDataSets, full SQLite SQL dialect support, ad-hoc TDataSet lookup	✓	✓

OTHER DATABASE TECHNOLOGIES

	TClientDataset for managing and manipulating datasets in-memory for Windows	✓	✓
	ClientDataSet support for 64-bit Linux		✓
	ClientDataSet support for 64-bit macOS, iOS, Android	✓	✓
	LiveBindings connect any type of data to any UI or graphical element in VCL and FMX	✓	✓
	dbGO™ for ADO connectivity for Windows (MDAC 2.8)	✓	✓
	InterBase Express (IBX) Components for Windows	✓	✓
	InterBase Express (IBX) Components support for iOS and Android	✓	✓

WEBBROKER AND WEBSTENCILS

ENHANCED IN 13 FLORENCE	WebBroker library for HTTP servers development and integration, supporting CGI, ISAPI libraries for Microsoft IIS, Apache HTTP server modules, and native HTTP servers based on Indy library. Extended in 13 with better Apache integration, Nginx web server support, and improved logging.	✓	✓
	WebBroker deployment support in Linux, as stand-alone or Apache modules		✓
INTRODUCED IN 11.1 ALEXANDRIA	WebBroker deployment support for Android	✓	✓
NEW IN 13 FLORENCE	WebBroker sessions management	✓	✓
ENHANCED IN 13 FLORENCE	WebStencils server-side script-based integration and processing of HTML files, to develop modern websites based on any JavaScript library, powered by the data extracted and processed by a RAD Studio server-side application. This is a sophisticated architecture introduced in RAD Studio 12.2, significantly extended in 13 with support for session data, global variables, a new switch statement, integrated WebBroker sessions authentication and authorization, direct access to dataset properties.	✓	✓
ENHANCED IN 13 FLORENCE	Use the WebStencils components, TWebStencilsProcessor and TWebStencilsEngine, to customize the template engine, control the file locations and the URL mapping, define URL parameters, and pass objects, datasets and collections to the scripts. The TWebStencilProcessor component implements the same interface of the old TPageProducer used in WebBroker and DataSnap and can replace it	✓	✓

RAD SERVER [3]

ENHANCED IN 13 FLORENCE	RAD Server turnkey REST based middleware stack that includes API hosting, data access, and SQL Database access, with significantly optimized performance [3]	✓
INTRODUCED IN 11 ALEXANDRIA	Freely, unlimited distributable*, limited-bandwidth, easy-to-deploy RAD Server Lite version (based on embedded InterBase ToGo database) **Requires a free to redeem license key*	✓
NEW IN 13 FLORENCE	Add support for server API versioning (that is, having a server offering multiple versions of the same REST APIs for older and newer client applications)	✓
INTRODUCED IN 12.3 ATHENS	The RAD Server engine is built with the FastMM5 memory manager, which improves performance, specifically in heavy multithreading scenarios.	✓
INTRODUCED IN 12.3 ATHENS	WebStencils integration in RAD Server by combining the existing TEMSFileResource component with a TWebStencilsEngine component	✓
	Build, test and deploy RAD Server packages on Linux	✓
	Create custom REST APIs for your business functionality and map them to custom URIs, using loadable packages for RAD server	✓
	Integrated with FireDAC high-performance enterprise data access for Oracle, DB2, Microsoft SQL Server, Informix, SQL Server, and many more database servers	✓
	EndpointProduce attribute to map MIME types (from Accept HTTP request header) to GET endpoints; EndpointConsume attribute to map MIME types (from Content-Type HTTP request header) to PUT, POST, PATCH endpoints	✓

[3] *Enterprise editions include free RAD Server Lite (RSLite) deployment and a single site RAD Server deployment license, while Architect editions include a multi-site deployment license*

INTRODUCED IN 12.3 ATHENS	Ability to easily support a smart ID model (squids), using a standard library to define better identifiers for resources exposed by the server.	
	HTTP Verb to Custom Method Name Mapping	
	Ability to Delegate Processing of a Request to a Custom Class or Component (sharing response processing code)	
	TEMSFileResource and TEMSDataSetResource components for an extremely simplified implementation of RAD Server endpoints mapped to file system folders access and database tables and queries access, including support for paging and sorting	
INTRODUCED IN 11 ALEXANDRIA	TRESTRequestDataSetAdapter component to simplify uploading a TDataSet (like a TFDMemTable) to a server via JSON -- for use with the TRESTResponseDataSetAdapter component on the server side	
	FireDAC database connection wizard, for generating a properly configured TEMSDataSetResource components mapped to queries for one or more selected database tables	
ENHANCED IN 13 FLORENCE	RAD Server (EMS) multi-tenancy support and console app for managing tenants configuration (on Windows and Linux). Version 13 adds to the RAD Server Multi-tenancy support an option to integrate tenant-specific configuration	
	RAD Server File Dispatching and multipart/form-data support	
	RAD Server Console UI redesign and migration to the Ext JS library	
	EMS Push Notifications server support for iOS and Android (updated to Firebase)	
	EMSClientAPI component to simplify EMS client side development	
	User, groups, sessions, and API calls analytics and reporting using a web based interface using the EMS Console (eventually filtered by tenant)	
	Available RAD Server Console (RSConsole.exe) client application to manage user accounts, edit the local configuration and run REST debugger-based requests against a RAD Server instance	
	ThingPoint provides developers with an enterprise access point between remote gadgets and devices; a ThingPoint can locally store, filter, and compute vast amounts of IoT data collected at the edge, while ensuring only critical data is synched with the central repository, a RAD Server	
ENHANCED IN 12.2 ATHENS	Support for declaring and retrieving EMS metadata, based on the Swagger open API initiative (http://swagger.io). This allows the use of the metadata language as YAML or JSON for the representation of the EMS REST API. Initial documentation generation is optional on the RAD Server module wizard. 12.2 adds support for custom metadata in Swagger and API filtering.	
	Installers to deploy the RAD Server engine and all of the required files to Windows and Linux servers (Delphi only)* ** Available for download in the GetIt Package Manager*	
	Support to deploy Delphi Linux applications in general and RAD Server applications in particular to Docker. ** Available for download in the GetIt Package Manager*	

DATASNAP MULTI-TIER

	Create multi-tier database applications with DataSnap for Windows servers	
	Create multi-tier database applications with DataSnap for Linux servers	
	Deploy DataSnap servers on Microsoft IIS server (HTTP or HTTPS) using the WebBroker architecture	
	Deploy DataSnap servers on Apache HTTP server (using HTTP or HTTPS) using the WebBroker architecture. For Delphi available also on Linux.	
	DataSnap clients uses System.NET for HTTP and HTTPS, with no need to deploy the OpenSLL client library (both for REST connections and HTTP connections)	
	DataSnap server methods support TDBXJSONStream parameter for passing JSON streams between client and server	
INTRODUCED IN 11 ALEXANDRIA	The REST URI is configurable using a mechanism, based on a TDSMethodMapEvent event	

	HTTPS support in stand-alone DataSnap Applications		
	DataSnap server ability to terminate socket connection. Communication Timeout for HTTP protocol		
	Specific DataSnap support for exposing FireDAC datasets and reconciling changes using deltas (with the unit FireDACJSONReflect), with data compression for better throughput optimization		
	Heavyweight callbacks support broadcasting to specific callbacks. Callback channel events for servers and clients. DataSnap REST server support for multiple Callback Tunnels		
	Authentication and role-based authorization.		
	Session Events for TCP/IP Protocol, Object support in DataSnap sessions		
	Monitor and control connections		
	DataSnap REST client connection (TDSRestConnection), for all platforms, with SecureProtocols property		
	Allow query parameters in REST calls and store them		
	Proxy information support for HTTP connections		
	Session events for increased simplicity of a multi-tier implementation		
	DataSnap wizards from the Object Gallery to make server and client creation even easier. DataSnap wizard source code enabling developers to build their own DataSnap server wizards		
	Added support for REST(ful) interface and exposure from the Server and REST client proxy generation		
	DataSnap Server Proxy generation for Object Pascal and C++ clients		
	Support for HTTP and HTTPS communication protocol beyond the local DataSnap (TCP/IP) approach; Added HTTP tunneling support for client code outside the firewall		
	Added Filter support for encryption and compression over-the-wire		
	Included encryption PC1 and compression filters included for quick enablement.		
	Lightweight Callbacks		
	Windows 32-bit and 64-bit native (DBX) clients and REST clients		
	macOS REST clients		
	REST / JavaScript client support for browser platforms		

INCLUDED DATABASE TOOLS AND DATABASE ENGINES

	Data Explorer support for FireDAC, optimized to add and manage database connections and browse database schema, including primary keys, foreign keys and sequences or generators		
	Data Explorer optimized to add and manage database connections and browse DB schema		
	Drag and Drop from FireDAC nodes and dbExpress nodes in the Data Explorer into forms and data modules to speed development		
	Direct SQL manipulation for any FireDAC connection		
ENHANCED IN 13 FLORENCE	InterBase 2020 Update 6 Developer Edition. Includes a license for the upcoming InterBase 15 Developer edition		
ENHANCED IN 12.2 ATHENS	IBLite 2020 Update 6 for Windows with unlimited deployment license		
ENHANCED IN 12.2 ATHENS	IBLite 2020 Update 6 for macOS, iOS, and Android, with unlimited deployment license		
ENHANCED IN 12.2 ATHENS	Free InterBase ToGo Deployment License for mobile platforms. A yearly subscription for unlimited deployment across desktop and mobile is available as a separate purchase.		

ENHANCED IN 12.2 ATHENS	IBConsole, the integrated InterBase configuration and management tool, recently updated to HighDPI and with an improved UI		

CLOUD SUPPORT

	Windows Azure components including AzureConnectionString, AzureBlobManagement, AzureQueueManagment, and AzureTableManagment		
	Ability to manipulate Windows Azure blobs, queues, and tables		
	MetaData support for TAzureQueueManagement		
	ContinuationToken support for Azure Table		
	Azure support updated to the current Azure APIs		
	Amazon Simple Storage Service (S3) API		
	Amazon Queue Service API		
	Amazon SimpleDB API		
	AWS support updated to the current AWS APIs and a more flexible support for regions		
	Deploy to Amazon EC2 and Windows Azure		

REST CLIENT LIBRARY AND BAAS

	REST Client Library for simplified invocation of REST services, with improvements in MIME types handling, properties (RedirectsWithGET and SecurityProtocols) and events (OnNeedClientCert and OnAuthEvent)		
ENHANCED IN 11.3 ALEXANDRIA	Authorization support including Basic Authentication, Plan Authentication, OAuth1, OAuth2. OAuth2 enhanced with automatic token refresh		
	TRestClient, TRestRequest, and TRestResponse components		
ENHANCED IN 12.1 ATHENS	REST Debugger tool for testing REST calls and their parameters. Offers an option to copy values from the data grid preview.		
	Enhanced REST Request parameters, with support for list parameters, parameters streaming, query parameters for all request methods, content body access		
	JSON reader and writers used in the REST client library		
	Backend as a Service (BAAS) architecture, with unified interfaces for the most common operations like user management, file storage, objects storage, and notifications support		
	Integrated BaaS support for Kinvey and the Parse API		

XML AND SOAP

	TXMLDocument component working on all platforms		
	TXMLDocument component support for MSXML on Windows; native OmniXML library and native ADOM library on other platforms		
	Easier selection of the default XML engine for XML data processing		
	Easily create Win32, Win64 SOAP Web Services		
	Build client-side SOAP Web Services, including support for SOAP 1.2 clients		
	SOAP clients use the HTTP client library, with full support for the native SSL layer on each platform (with improved support for client certificates)		
	WSDL External Schema support		
	XML Transformation Tools and Components to exchange data between XML and datasets		
	Native Object Pascal and C++ XML bindings to simplify XML programming – access XML documents using interfaces		

	SOAP runtime support for optional and unbounded elements		

FMX Linux

ENHANCED IN 13 FLORENCE	FireMonkey for Linux UI library (FMX Linux), bringing any FireMonkey application to the Linux desktop platform* ** Available for download in the GetIt Package Manager (not included in Community Edition)*	
ENHANCED IN 13 FLORENCE	New enhanced integration with the RAD Studio IDE for FMX Linux	

ADDITIONAL AND THIRD PARTY INTEGRATED TOOLS

FILE COMPARE WITH BEYOND COMPARE TEXT COMPARE

	Beyond Compare Text Compare integration		
	Compare and edit the contents of files with syntax highlighting, ignore changes in whitespace and comments		
	Print reports of differences or save them as HTML or plain		
	Filter display to only show differences, optionally with a few lines of context		
	Built-in support for Object Pascal, C++, HTML, DFMs, and more		
	Adds a "Compare" menu to the IDE's "Edit" menu and Project Manager, with commands to compare against previous revisions, other editor tabs, and original files		
	Automatically configured for both differences and merges from the History view		

LOGGING WITH CODESITE EXPRESS

ENHANCED IN 11.2 ALEXANDRIA	Raize Software CodeSite Express integration provides advanced application logging and debugging capabilities* ** Available for download in the GetIt Package Manager (not included in Community Edition)*		
	Log all kinds of information without data conversions including strings, numbers, dates & times and even objects, string lists, exceptions, and much more		
	Log information from application code (including recursive functions and multiple threads) without interrupting program execution or causing side effects		
	Emphasize important logging information through a variety of message types		
	Send logging messages to a CodeSite Log File for later review or to the CodeSite Live Viewer for real-time analysis, or to both simultaneously		
	Utilize the extensive analysis tools in the CodeSite Viewers to analyze message logs and locate problem areas faster		
	Filter message logs by application name, process id, thread name, computer name, category, and message text		
	Quickly organize the message log into multiple views by using the advanced CodeSite Message Organizer		
	Control the amount of logging information generated by using additional CodeSite loggers with their own Category instead of relying on arbitrary logging levels		

ADDITIONAL COMPONENTS

UPDATED IN 11.3 ALEXANDRIA	TeeChart Standard components including chart, DBchart, and a rich set of different business graph types.		
ENHANCED IN 13 FLORENCE	In addition, we offer a number of exclusive components in the GetIt Package manager, for Dephi and / or C++. You can check https://getitnow.embarcadero.com/ to see what is currently available (some of the exclusive items might be limited to higher end SKUs)		

SENCHA EXT JS PROFESSIONAL EDITION [4]

Ext JS HTML5/JavaScript (Framework and UI Component Library) – Pro Edition	ARCHITECT ONLY [4]
Cmd: Build Optimization Tool	ARCHITECT ONLY [4]
Ext JS Stencils: Design Kit	ARCHITECT ONLY [4]
Architect: Visual App Builder	ARCHITECT ONLY [4]
Sencha Themer: Styling Tool	ARCHITECT ONLY [4]

4 Ext JS Web Development Framework and Aqua Data Studio are not included in Architect LC edition

AQUA DATA STUDIO [4]

Register Servers/Databases, Database Navigator/Explorer	ARCHITECT ONLY [4]
Query Analyzer (SQL editor) and Query Builder	ARCHITECT ONLY [4]
Table Data Editor	ARCHITECT ONLY [4]
Visual Analytics	ARCHITECT ONLY [4]
Schema/Data/File Compare Import/Export Data and DDL	ARCHITECT ONLY [4]
ER Modeler	ARCHITECT ONLY [4]
Database Administration	ARCHITECT ONLY [4]

4 Ext JS Web Development Framework and Aqua Data Studio are not included in Architect LC edition

Download a Free Trial at www.embarcadero.com/products

CORPORATE HEADQUARTERS | EMBARCADERO TECHNOLOGIES, INC. 4001 W. Parmer Lane, Suite 125, Austin, TX 78727, USA
www.embarcadero.com | sales@embarcadero.com

© 2025 Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos, and all other Embarcadero Technologies product or service names are trademarks or registered trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners. 250915