

RAD STUDIO 11.1 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. Community edition¹ is available only for single personalities and has a limited license. Some of the features are only available for download from the IDE in the GetIt Package Manager, requiring an Internet connection.












































































































TABLE OF CONTENTS













































































































INTEGRATED COMPILERS AND TOOLCHAINS	APPLICATION PLATFORMS, INTEGRATED FRAMEWORKS, DESIGNERS AND SDKS
LICENSING AND TOOL MANAGEABILITY	VISUAL COMPONENT LIBRARY (VCL)
FMX APPLICATION PLATFORM (FIREMONKEY)	INTERNET OF THINGS (IOT) SUPPORT
COM/DCOM SUPPORT	INTEGRATED BUILD TOOLS AND PROJECT SUPPORT
VISUAL LIVEBINDINGS	RECENT DELPHI LANGUAGE AND CORE RTL FEATURES
RECENT C++ LANGUAGE AND CORE RTL FEATURES	INTEGRATED DEVELOPMENT ENVIRONMENT AND DEVELOPER PRODUCTIVITY
SHARED RUNTIME LIBRARY FEATURES	INTERNATIONAL DEVELOPMENT
SOURCE CODE MANAGEMENT	UML MODELING
REFACTORINGS	INTEGRATED UNIT TESTING
INTEGRATED DEBUGGING	DATABASE APPLICATION DEVELOPMENT AND CONNECTIVITY
INTEGRATED HELP	DBEXPRESS AND IBX
FIREDAC MULTI-DEVICE DATA ACCESS LIBRARY	DATASNAP MULTI-TIER
RAD SERVER	INCLUDED DATABASES
DATABASE TOOLS	CLOUD SUPPORT
REST CLIENT LIBRARY AND BAAS	IP*WORKS! COMMUNICATION COMPONENTS
XML AND SOAP	FILE COMPARE WITH BEYOND COMPARE TEXT COMPARE
CONNECT TO REST APIS WITH ENTERPRISE CONNECTORS	LOGGING WITH CODESITE EXPRESS
TEECHART CHARTING COMPONENTS	SENCHA EXT JS PROFESSIONAL EDITION
FASTREPORT REPORTING SOLUTIONS	
AQUA DATA STUDIO	





















































































































FEATURE		PROFESSIONAL / COMMUNITY ¹	ARCHITECT / ENTERPRISE
INTEGRATED COMPILERS AND TOOLCHAINS			
	Delphi 32-bit native code optimizing compiler (dcc32) and toolchain for Windows.		
	Delphi 64-bit native compiler and toolchain for Windows (dcc64)		
	Delphi 64-bit native compiler and toolchain for Linux (dcclinux64)		
INTRODUCED IN 10.3.2 RIO	Delphi native compiler and toolchain for macOS 64-bit (dccosx64)		
NEW IN 11 ALEXANDRIA	Delphi native compiler and toolchain for macOS ARM 64-bit (dccosxarm64)		
	Delphi compiler and toolchain for iOS 64 bit devices (dcciosarm64)		
	Delphi ARM 32-bit compiler and toolchain for Android devices (dccaarm)		
INTRODUCED IN 10.3.3 RIO	Delphi 64-bit ARM compiler and toolchain for Android 64-bit platform (dccaarm64)		
INTRODUCED IN 10.3 RIO	C++ 17 Clang-enhanced compiler and toolchain for Win32 (bcc32x and bcc32c). Classic Embarcadero C++ compiler and toolchain for Win32 (bcc32)		
INTRODUCED IN 10.3.2 RIO	C++ 17 Clang-enhanced compiler and toolchain for Win64 (bcc64)		
	C++Builder iOS 64-bit C++ compiler and toolchain		
	C++Builder Android 32-bit ARM compiler and toolchain (bccarm)		
LICENSING			
INTRODUCED IN 10.2.3 TOKYO	Community Edition with limited license based on revenue and team size (see EULA for details)	 	
		COMMUNITY only	
	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	

¹ Community edition planned for a future 11.x release. Current available Community Edition version is 10.4.2. Community Edition doesn't include some of the bonus features available in the GetIt package manager.

	Network named or concurrent licenses available	 	 
		PROFESSIONAL only	
ENHANCED IN 11 ALEXANDRIA	Earlier version licenses included in network licenses (Delphi 2007–10.4, C++ Builder 2007–10.4). Delphi 7 and C++Builder 6 available separately	 	 
		PROFESSIONAL only	
ENHANCED IN 11 ALEXANDRIA	Access to licenses and downloads for earlier versions with purchase of standalone product licenses (Delphi 2007–10.4, C++Builder 2007–10.4, Delphi 7, and C++Builder 6)	 	 
		PROFESSIONAL only	
APPLICATION PLATFORMS, INTEGRATED FRAMEWORKS, DESIGNERS AND SDKS			
ENHANCED IN 11 ALEXANDRIA	Support for Linux server-side and standalone applications for Ubuntu 20.04 LTS and 18.04 LTS, RedHat Enterprise Linux v8, WSL 2 (Windows Subsystem for Linux)		
ENHANCED IN 11.1 ALEXANDRIA	Support for Linux client-side applications using the bundled FMX Linux library* * Available for download in the GetIt Package Manager		
ENHANCED IN 11.1 ALEXANDRIA	FireMonkey Application Platform for creating 32-bit Windows applications and 64-bit Windows applications for Windows 11, Windows 10, Windows 7 (SP1+); Windows Server 2019 and 2016	 	 
ENHANCED IN 11.1 ALEXANDRIA	FireMonkey Application Platform for creating 64-bit macOS applications for Intel and ARM CPUs (M1), that run on macOS 12 Monterey, 11 Big Sur and 10.15 Catalina, including support for universal binary		
ENHANCED IN 11.1 ALEXANDRIA	FireMonkey Application Platform for creating iOS ARM 64-bit applications for iOS 15 and iOS 14	 	 
ENHANCED IN 11.1 ALEXANDRIA	FireMonkey Application Platform for creating native Android ARMv7 applications for Android 12, 11, 10, Pie (9.0), Oreo (8.1)	 	 
ENHANCED IN 11.1 ALEXANDRIA	FireMonkey Application Platform for creating native 64-bit Android ARMv7 applications for Android 12, 11, 10, Pie (9.0), Oreo (8.1)		
ENHANCED IN 11.1 ALEXANDRIA	VCL (Visual Component Library) for rapidly building 32-bit and 64-bit applications for Windows 11, Windows 10, Windows 7; Windows Server 2019 and 2016	 	 
ENHANCED IN 11 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. <i>(This feature requires running RAD Studio on Windows 10 Anniversary Update or above)</i>	 	 
ENHANCED IN 11.1 ALEXANDRIA	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)	 	 
FMX APPLICATION PLATFORM (FIREMONKEY)			
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	 	 
	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 11.1 ALEXANDRIA	FireMonkey for Windows now 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	 	 
INTRODUCED IN 10.4 SYDNEY	Metal Drivers GPU Support for macOS		
INTRODUCED IN 10.4 SYDNEY	Metal Drivers GPU Support for iOS	 	 
INTRODUCED IN 10.4 SYDNEY	Windows Styled Memo Implementation with enhanced IME support	 	 
INTRODUCED IN 10.4 SYDNEY	TBufferedLayout Component for faster rendering of child controls	 	 
INTRODUCED IN 10.4 SYDNEY	Support for iOS Launch Storyboard	 	 

INTRODUCED IN 10.4 SYDNEY	iOS Theme check and theme change 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	 	 
INTRODUCED IN 10.3.2 RIO	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 now 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	 	 
INTRODUCED IN 10.3 RIO	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. Now with 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	  	 
	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	 	 
NEW IN 11 ALEXANDRIA	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)	 	 
	WebBrowser component for Windows, with support for both classic the IE ActiveX and the new Microsoft WebView 2 control (Chromium-based Edge)	 	 
INTRODUCED IN 10.4 SYDNEY	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 more 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 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	 	 
INTRODUCED IN 10.4.2 SYDNEY	Advertising component for iOS, including AdMob support based on the Google Firebase SDK	 	 
ENHANCED IN 10.4.2 SYDNEY	Push notifications support for mobile platforms, to receive remote notifications even if the app is not running (updated to include 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	 	 
	Phone Dialer Support for iOS and Android	 	 
	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 now 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 (now 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, 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 applications 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	 	 
	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, password control and more	 	 
	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	 	 
INTRODUCED IN 10.3.1 RIO	5 FireMonkey multi-device styles* with theming support for all supported platforms <i>* Available for download in the GetIt Package Manager</i>	 1  1	 
	Bitmap Style Designer enhancement, including templates for Windows, Mac OS and Android Lollipop styles, ability to create blank styles for different platforms, and to adjust objects for different graphic resolutions	 	 
	Significantly improved integrated FireMonkey Style Designer, and enhancement to the TStyleBook component to support a collection of styles for various platforms	 	 
	StyleViewer for Windows 10 Style in Bitmap Style Designer	 	 
	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	 	 
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 10.4 SYDNEY	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	 	 
INTRODUCED IN 10.4 SYDNEY	A dozen of ready to use High-DPI optimized VCL styles	 	 
INTRODUCED IN 10.4 SYDNEY	Per-control VCL styling support	 	 
INTRODUCED IN 10.4 SYDNEY	Custom Title Bars and the TitleBarPanel Component	 	 
INTRODUCED IN 10.4 SYDNEY	TVirtualImage component with support for images with multiple resolutions	 	 
ENHANCED IN 10.4.2 SYDNEY	TEdgeBrowser Component integrating Microsoft's Chromium-based Edge WebView2 component and changes to the TWebBrowser component to support both IE and Edge-- now based on Microsoft release version of Edge WebView2 component and supporting custom cache folders	 	 
INTRODUCED IN 10.4.2 SYDNEY	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	 	 
INTRODUCED IN 10.4.2 SYDNEY	VCL TNumberBox control, a modern-looking numeric input control, for integer, floating point numbers, and currency values	 	 
	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	 	 
	Windows components fully integrated with the classic Windows API and the modern WinRT API	 	 
INTRODUCED IN 10.3 RIO	Windows API headers for areas like High-DPI and DPI-awareness, WM_POINTER pen input support, and more	 	 
	Complete VCL library source code to help writing custom components, debug your applications, and fully understand the library behavior	  Limited in COMMUNITY	 
INTRODUCED IN 10.3 RIO	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.	 	 
INTRODUCED IN 10.3 RIO	Support for Windows 10 "Per Monitor V2" mode in application manifest and enhanced system metrics API support (requires Windows 10 Creator's Update, build 1703)	 	 
ENHANCED IN 10.3 RIO	High-DPI Awareness and 4K monitor support, plus Windows 8.1/10 multi-monitor support for VCL applications, with dozens of improvements	 	 
	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	 	 
ENHANCED IN 10.3 RIO	VCL Font Scaling improvements for higher DPI and higher resolution monitors	 	 
ENHANCED IN 10.4.2 SYDNEY	Updated Konopka Signature VCL Controls (over 200 additional Windows UI controls)* <i>* Available for download in the GetIt Package Manager (excluding CE)</i>	  	 
	VCL sensor components for Windows tablets	 	 
	VCL support for AppTethering across desktop and mobile applications, 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	 	 
INTRODUCED IN 10.3.1 RIO	10 VCL Windows styles* that users can apply to their Windows applications <i>* Available for download in the GetIt Package Manager</i>	 1 	 
	Create your own custom styles with the VCL Style Designer	 	 
	VCL Styling improvements, including support for styling common dialogs and the TWebBrowser component	 	 
NEW 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, TToolBar, TNumberBox and other components.	 	 
NEW IN 11 ALEXANDRIA	TTreeView support for CheckBoxes	 	 
NEW IN 11 ALEXANDRIA	New 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)	 	 
	TTouchKeyboard a virtual keyboard for enhanced non-keyboard interface interactions that support multiple locales and languages	 	 
	Modern VCL Components: TCategoryPanelGroup, TButtonEdit, 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. Now with 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, buttonedit, 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	 	 
ENHANCED IN 10.4 SYDNEY	TWebBrowser support, wrapped on the Microsoft IE ActiveX component (now 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	 	 
INTERNET OF THINGS (IOT) SUPPORT			
	Native Bluetooth APIs (on platforms that fully support the technology) for FireMonkey and VCL applications	 	 
	Bluetooth framework improvements and TBluetooth component for classic Bluetooth	 	 
	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	 	 
	TBeaconDevice class for turning a device on one of the supported platforms into a “beacon”	 	 
	BeaconFence components* for detecting a device position relative to predetermined zones based and based on triangulation of beacons positions <i>* Available for download in the GetIt Package Manager</i>		 
	BeaconFence enhancements for detecting zones, Windows support, customized calculation of the current position, access last calculated position	 	 
VISUAL LIVEBINDINGS			
INTRODUCED IN 10.4 SYDNEY	Optimized Visual Live Bindings RTTI integration for better performance	 	 
INTRODUCED IN 10.4 SYDNEY	VCL and FMX controls bound to a TDataSet and TField now respect some of the TField properties like Alignment, EditMask, DisplayWidth	 	 
ENHANCED IN 10.3 RIO	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 from 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 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		
	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	 	 
ENHANCED IN 10.4.2 SYDNEY	IDE support for creating Android App Bundle (AAB) including both 32-bit and 64-bit binaries for store submission		
INTRODUCED IN 10.4 SYDNEY	Integrated installation of AdoptOpenJDK, required by Android SDK tools	 	 
	IDE support for adding Java classes to an Android app classes.dex library and customizing it	 	 
	Support for Android Services in the IDE, including wizards to create Android Services and to add them to an existing Android app		
	Java2OP (Java to Object Pascal bridge) API headers translation tool (for Android), improved to reduce dependency between units		
	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), now with several iOS device provisioning improvements to simplify the deployment process – including Auto Bundle identifier and support for executing an application from the IDE using the AdHoc target	 	 
	Define iOS application splash screens and mobile app icons in the IDE	 	 
	Deploy apps to Android devices (Debug/App Store)	 	 
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 Mac macOS tray app that helps managing multiple versions of Platform Assistant (PAServer)	 	 
	Deployment Manager: Deploy to iOS, Android, Windows	 	 
	Deployment Manager: Deploy to 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	 	 
	Delphi 1 through 10.4 project import and C++Builder 3 through 10.4 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, OBJ, and LIB files, GREP tool, and MAKE tool	 	 
RECENT C++ LANGUAGE AND CORE RTL FEATURES			
ENHANCED IN 11 ALEXANDRIA	Win32 C++ Clang-enhanced compiler (with C++ 17 language support)		
ENHANCED IN 11 ALEXANDRIA	Win64 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 10.3 RIO	Improved C++ and Delphi ABI and header compatibility	 	 
NEW 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, Win64, iOS32, iOS64, and Android, including deployment support		
	Improved optimizations for C++ Clang-enhanced compilers code generation (-O1 and -O2 and now also -O3 optimizations)		
	Simplified array initialization and construction for C++ Clang-enhanced compilers		
	Android (32-bit) and iOS32 compilers updated to Clang version 3.3		
	The Clang-enhanced compilers are now 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		
	ANSI/ISO C++11 and 99 Standard language conformance support for 64-bit		
	#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		
NEW 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 11 ALEXANDRIA	Binary literals and digit separator support. Inline assembler support for AVX instructions (AVX-512)		
INTRODUCED IN 10.4 SYDNEY	Custom managed records with the ability to define custom Initialize, Finalize and Assign operations, executed when a record is allocated, de-allocated, and copied		

INTRODUCED IN 10.4 SYDNEY	Standard functions for low-level bit processing operations: CountLeadingZeros, CountTrailingZeros, CountPopulation		
INTRODUCED IN 10.3 RIO	Inline variables declaration, supporting local scope, direct initialization, and type inference; inline constants declaration		
INTRODUCED IN 10.3 RIO	Type inference for inline variable declarations (the type is inferred from the value assigned to the variable)		
ENHANCED IN 10.3 RIO	Support for calling WinRT APIs, with now over 50 specific Object Pascal interface units		
	[Weak] and [Unsafe] support for interface references		
	[Volatile] attribute can be attached to a method parameters and local variables		
	Delphi compiler and linker performance improvements to improve compilation speed for generic methods and generic methods in standard classes		
	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. Now 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			
INTRODUCED IN 10.4 SYDNEY	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	 	 
INTRODUCED IN 10.3 RIO	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 <i>contract sources</i>) 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	 	 
ENHANCED IN 10.4 SYDNEY	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 and C++ RTL for iOS, iOS 64-bit API headers and RTL integration	 	 












































































INTRODUCED IN 10.3.3 RIO	Delphi RTL for 64-bit Android		
ENHANCED IN 11 ALEXANDRIA	Delphi and C++ RTL for Android, with support for Android API level 30. Additionally, we added support for the new "AndroidX" libraries	 	 
	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 now has a callback to use during the extraction process to know the current progress, support added to extract ZIP files with passwords. Now with Zip64 support and additional features.	 	 
ENHANCED IN 10.3 RIO	Regular Expressions (Regex) library, now supporting PCRE UTF-16 on Windows	 	 
	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	 	 
	Box2D physics engine, for manipulating on-screen objects according to the physics rules	 	 
	Object Pascal interfaces for the Box2D physics engine		
ENHANCED IN 10.3 RIO	TStringBuilder for easier and faster string concatenation, optimized and with flexible growth strategy	 	 
ENHANCED IN 10.3 RIO	JSON (JavaScript Object Notation) processing using a JSON.NET implementation for JSON streaming with readers and writers (including base TJsonReader and TJsonWriter classes)	 	 
ENHANCED IN 10.3 RIO	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)	 	 
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, now 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 OpenSSL library), with improved asynchronous support	 	 
ENHANCED IN 11 ALEXANDRIA	NetHttpClient and NetHttpRequest components, offering easy access to the HTTP client framework, now with HTTP / 2 protocol version support	 	 
	Support for native HTTP(S) client libraries on Windows, macOS (Delphi only), 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 AND DEVELOPER PRODUCTIVITY			
ENHANCED IN 11.1 ALEXANDRIA	The RAD Studio IDE is now high-DPI enabled (using perMonitor v2 Windows configuration). All primary forms and panes have been enhanced, with new 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 11.1 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.	 	 
NEW IN 11 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 disabled by default.	 	 
ENHANCED IN 11 ALEXANDRIA	Edit the code for and design a form for the same form at the same time in multiple windows, including moving the designer between windows	 	 
ENHANCED IN 11.1 ALEXANDRIA	Completely redesigned and VCL-based Welcome Page, offering 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 new ToolsAPI.	 	 
NEW IN 11 ALEXANDRIA	New FireMonkey designers guidelines and designer hints (component hints and position hints)	 	 
ENHANCED IN 11.1 ALEXANDRIA	New Delphi Code Insight implementation (based on the Language Server Protocol architecture) 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. The classic engine has been removed.		
ENHANCED IN 11.1 ALEXANDRIA	C++ Code Insight implementation (based on the Language Server Protocol architecture and cquery) has been significantly improved in terms of quality and performance.		
NEW IN 11 ALEXANDRIA	New C++ code formatter based on clang-format, including the ability to format as you type and several predefined formatting styles		
NEW IN 11 ALEXANDRIA	Support for using Delphi LSP in Visual Studio Code and other external editors	 	 
INTRODUCED IN 10.4 SYDNEY	Code editor status bar font zooming, syntax highlighter and file encoding selections	 	 
INTRODUCED IN 10.4 SYDNEY	Implementation of the IDE Title bar (using the matching VCL control)	 	 
INTRODUCED IN 10.3.1 RIO	IDE productivity tool: Bookmarks*, which extends the IDE's previous marking of locations in the code editor. <i>* Available for download in the GetIt Package Manager (not for Community Edition)</i>	  1	 
INTRODUCED IN 10.3.1 RIO	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. <i>* Available for download in the GetIt Package Manager (not for Community Edition)</i>	  1	 
INTRODUCED IN 10.3 RIO	IDE main window UX extensively improved, with a clean and modern look and feel designed for long-term use and productivity	 	 
INTRODUCED IN 10.3 RIO	Commonly used dialogs (Project Options, IDE Options, New Items dialog, GetIt Package Manager dialog, Compiler progress dialog) have been redesigned and improved, for productivity and clarity	 	 
INTRODUCED IN 10.3 RIO	Search added to the Project Options and IDE Options dialogs, allowing you to search for a setting inside each dialog	 	 
ENHANCED IN 10.3 RIO	Styled IDE UI, including ability to toggle and disable IDE styling and support for a dark style. In 10.3 the IDE has a Light theme based around the blue color and an updated Dark theme	 	 
ENHANCED IN 10.3 RIO	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	 	 
NEW 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		
	The C++ Structure Parser has been rewritten and includes support for modern C++		







































































	Method Toxicity Metrics; Bookmarks stack, Selection Expansion		
	Modular View menu, re-arranged with subsections	 	 
	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	 	 
	Redesigned compiler unit caching mechanism, to free older units in case the available memory is being filled up, when compiling project groups with multiple projects	 	 
	Improved 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	 	 
	Enhanced IDE Project Options to easily enable High-DPI Awareness in your applications, now supporting also 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	 	 
	GetIt Library Manager for easy discovery, download and update of source code libraries, components, and other features from Embarcadero GetIt servers*. Extended with Categories, UI improvements, and actions * Use of GetIt requires Internet connectivity. The feature is not available to off-line developers.	 	 
ENHANCED IN 11.1 ALEXANDRIA	GetIt Library Manager enhancements with support for updates, sort by date, subscription only packages, cached images, and an updated UI.	 	 
ENHANCED IN 10.4.2 SYDNEY	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	 	 
ENHANCED IN 10.3.2 RIO	Project statistics information for activity tracking and to better understand team productivity (now optional and disabled by default)	 	 
	Multi-paste support to let you perform the same paste operation on multiple source code lines at once	 	 
	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		
	Project Manager enhancements, with the support for adding database files and simplifying their deployment	 	 
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). New 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	 	 
ENHANCED IN 10.3 RIO	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	Code Formatter and beautification for source code including block formatting	 	 
ENHANCED IN 11 ALEXANDRIA	Command-line Code Formatter for integration with automated build processes, enhanced in 11 Alexandria with clang-format support for C++	 	 
	Code Formatter profiles allow users to create / save / load custom formatter options sets		

NEW 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		
	Audits and metrics		
	Search for Usages command now available from the code editor and Delphi Class Explorer context menus		
	Compilation can be done in a background thread	 	 
INTRODUCED IN 10.4.2. SYDNEY	Support for C++ parallel building through batch compilation		
	Build a C++ project accelerated over multi-core with the TwineCompile add-on <i>* Available for download in the GetIt Package Manager (not for Community Edition)</i>	 1	
	Checkbox for Boolean types in the Object Inspector greatly enhancing readability	 	 
	Property Editors for Date properties now use Calendar controls	 	 
	Use Unit interaction capabilities increased		
	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 greatly simplifies 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	 	 
	Project Manager now has the ability to run selected applications with either Run or Run without Debug from the context menu.	 	 
	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	 	 
ENHANCED IN 10.3 RIO	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	 	 
	Symbol Insight – click to navigate to declarations and implementations	 	 
	Code Folding configuration and properties	 	 
	Editor Gutter – easier to read with less clutter	 	 
ENHANCED IN 10.3.2 RIO	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	 	 
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	 	 
Subversion/History integration	 	 
A full default Diff viewer	 	 
Git version control integration into the IDE, now 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	 	 
INTERNATIONAL DEVELOPMENT		
UnicodeString type ; Unicode-enabled development environment; Unicode-enabled VCL and FMX frameworks	 	 
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	 	 
REFACTORINGS		
Prototype synchronization – as you change the prototype for a function, you can use this feature to synchronize the interface and implementation sections to match (Object Pascal language only)		
Extract method and change parameters refactoring, including support for generic types		
Rename refactoring for Delphi		
Refactor driven Find References		
Other refactorings include: Introduce Variable, Introduce Field, Inline Variable, Safe Delete, Push Members Up / Down, Pull Members Up, Extract Superclass, Extract Interface, Move Members, Declare variable, Declare field, Find unit/import namespace, Extract to resource string		
Refactor driven “Find in Files”		
UML MODELING		
UML Code Visualization – at any time, get a UML model view of your source code		
Object Pascal code generation from class diagram		
Object Ancestry; show an object full hierarchy		
Automatic Sequence diagrams generation support		
Ability to load an image into the diagram for annotations and advanced documentation		
Sequence diagrams		
Collaboration diagrams		
State charts		
Deployment diagrams		
Use case diagrams		
Activity diagrams		
Component diagrams		
Documentation generation		

















	Run Documentation generation from the command-line		
	80+ Static Code Metrics that allow for deep understanding and analysis of the health of the object domain		
	Kiviat graph enhancements for interactive reporting		
	10+ introductory level static code Metrics for understanding the health of the object diagram		
	Run static Metrics from the command-line		
	10 introductory level static code audits for better code understanding		
	200+ Static Code Audits for deep code analysis to better understand the coding style and approach		
	Pointer Analysis for Audits. Run static Audits from the command-line		
	Import model from Rational® Rose (mdl import)		
	Language neutral UML 1.5 and 2.0 modeling		
	Transformation from language neutral UML 1.5 and UML 2.0 projects to source code projects		
	Design Patterns support, including GOF patterns		
	Diagram printing		
	XMI 1.1 Import/Export		
	Full two-way class diagrams with synchronization between code and diagrams		
	OCL 2.0 support		
	“UML in Color” profile		
	Diagram hyper-linking and annotations		
INTEGRATED DEBUGGING			
	Full-featured debugger with color syntax highlighting	 	 
	Debug remote and local 64-bit Windows applications	 	 
ENHANCED IN 11 ALEXANDRIA	New debugger for C++ for Win64, based on LLDB 9, including formatters to evaluate common complex types like UnicodeString and AnsiString, C++ strings, vectors, deque, and more		
ENHANCED IN 11.1 ALEXANDRIA	Debug remote macOS 64-bit applications (Intel and ARM). The Delphi macOS 64-bit ARM debugger is now based on the LLDB debugger architecture,		
	Remote iOS 64-bit device debugging	 	 
	Android device debugging	 	 
ENHANCED IN 11.1 ALEXANDRIA	Debug Android 64-bit apps on device. The Delphi Android 64-bit debugger is now based on the LLDB debugger architecture,		
	Remote Linux 64-bit debugging		
	Improved debugging support for Clang-enhanced C++ compilers for Win32		
	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 now 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		
	CPU view for low-level debugging		
	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			
ENHANCED IN 10.3 RIO	Integrated unit testing with the DUnitX testing framework		
ENHANCED IN 10.3 RIO	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			
	Support for CHM help engine and updated IDE help structure	 	 
	Continually updated help system with diagrams showing class hierarchy and declared members pages	 	 
DATABASE APPLICATION DEVELOPMENT AND CONNECTIVITY			
	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 from 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.	 	 
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 <i>Some features in this table marked as ² are limited to local connections in Professional and Community Editions</i>	 	 
	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.		 
	iOS and Android mobile database support, including SQLite, IBToGo and IBLite	 	 
ENHANCED IN 10.3 RIO	Updated FireDAC Microsoft SQL Server driver, supporting MS SQL 2017		 
ENHANCED IN 11.1 ALEXANDRIA	Updated FireDAC MySQL driver, supporting MySQL v 8.0 and MariaDB v 10.6, MySQL Prepared Statements API, internal BLOB streaming	 ²  ²	 
	Updated FireDAC ODBC driver 13, with support for data type mapping by data type name		 
ENHANCED IN 11.1 ALEXANDRIA	Updated FireDAC SQLite driver with support for both static and dynamic linkage of SQLite engine. Includes support for SQLite SEE.	 	 
ENHANCED IN 10.3 RIO	Updated 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 11 ALEXANDRIA	Updated 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.		 
	TFDOracleAdmin service component		 
	Updated 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	 	 
	Updated FireDAC DB2 driver, with support for DB2 on AS/400, IDENTITY columns, and a connection definition parameter		 










































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





























































ENHANCED IN 11.1 ALEXANDRIA	Updated FireDAC Firebird driver to look for the driver in VendorHome. Previous versions included support for version 3.0.4 and Firebird embedded, including local connection protocol, FB\$OUT package, support for long statements, little/big endian configuration in connection parameters, and support for new Firebird 4 data types.	 2  2	 
ENHANCED IN 11 ALEXANDRIA	FireDAC PostgreSQL driver enhancements for PostgreSQL up to version 13, including support for PostgreSQL Stored Procedures. Past updates included identity columns, macaddr8, password encryption and little/big endian configuration in connection parameters	 2  2	 
ENHANCED IN 10.3 RIO	FireDAC SQL Anywhere driver, now 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 10.3 RIO	Complete FireDAC support for the NoSQL MongoDB database, including a FireDAC MongoDB driver, now with TimeZone connection parameter		 
	MongoDB API wrapping classes, including TMongoConnection, TMongoDatabase, TMongoCollection and more		 
	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 Iterate 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, now with additional support for Oracle and MongoDB databases		 
	Improved FireDAC updates management	 	 
	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 now displayed in the Object Inspector as a record	 	 
	IBLite specific driver for desktop and mobile	 	 
	InterBase encryption connection options	 	 
	Windows/Mac 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	 	 
	Improvements of FireDAC cached updates mode, including UpdateOptions.AutoCommitUpdates property and improved 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, now 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		
DBEXPRESS AND IBX			
	dbExpress™ 4 database connectivity framework* with connection pooling, tracing, and delegate drivers <i>* The dbExpress technology is not being further extended and it's now considered deprecated, while we keep including it. We recommend migration to FireDAC</i>		
	dbExpress local connectivity to InterBase, MySQL, SQLite		
	Included dbExpress drivers available for 64-bit Windows – InterBase local, MySQL local, and SQLite local		
	Included dbExpress drivers available for 64-bit Windows – InterBase, Firebird, Oracle, MySQL, SQL Anywhere, and Informix		
	TSQLMonitor support		
	InterBase ToGo dbExpress driver for Windows, iOS and Android		
	dbExpress Server connectivity to InterBase® XE7 and XE3, Firebird 2.5, 2.1 and 1.5, Oracle® 11g and 10g, Microsoft SQL Server® 2008, 2005, and 2000, Informix® 9x (not Unicode-enabled), IBM DB2® 9.x (not Unicode-enabled), SQL Anywhere™ 12 and 11 (Unicode-enabled); and SQL Anywhere 9 (not Unicode-enabled), Sybase® 12.5 (not Unicode-enabled), MySQL 5.1 and 5.0 (Unicode enabled) and 4.1		
	dbExpress ODBC Driver		
	TClientDataSet for managing and manipulating datasets in-memory (with support for all platforms)		
	ClientDataSet support for 64-bit Linux		
INTRODUCED IN 10.3.2 RIO	ClientDataSet support for 64-bit macOS		
	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, including support for iOS and Android		

RAD SERVER ³


³ Enterprise editions include also a single site RAD Server deployment license (additional RAD Server deployment licenses are sold separately), while Architect editions include a multi-site deployment license







ENHANCED IN 10.3.2 RIO	RAD Server (formerly known as EMS) turnkey REST based middleware stack that includes API hosting, data access, and SQL Database access, with significantly optimized performance ³	 
NEW IN 11 ALEXANDRIA	Freely, unlimited distributable*, limited-bandwidth, easy-to-deploy RAD Server Lite version (based on embedded InterBase ToGo database) <i>*Requires a free to redeem license key</i>	 
	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	 
INTRODUCED IN 10.3 RIO	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	 
INTRODUCED IN 10.3 RIO	HTTP Verb to Custom Method Name Mapping	 
INTRODUCED IN 10.3 RIO	Ability to Delegate Processing of a Request to a Custom Class or Component (sharing response processing code)	 
INTRODUCED IN 10.3 RIO	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	 
NEW IN 11 ALEXANDRIA	New 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	 
INTRODUCED IN 10.3.2 RIO	FireDAC database connection wizard, for generating a properly configured TEMSDatasetResource components mapped to queries for one or more selected database tables	 
ENHANCED IN 11 ALEXANDRIA	RAD Server (EMS) multi-tenancy support and console app for managing tenants configuration (on Windows and Linux)	 
	RAD Server File Dispatching and multipart/form-data support	 
ENHANCED IN 10.3.2 RIO	RAD Server Console UI redesign and migration to the Ext JS library	 
ENHANCED IN 10.3.2 RIO	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)	 
ENHANCED IN 10.3.2 RIO	Redesigned and expanded 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, an Enterprise Mobility Services server	 
ENHANCED IN 10.3.2 RIO	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 now an option of the RAD Server module wizard.	 
INTRODUCED IN 10.3.2 RIO	Installers to deploy the RAD Server engine and all of the required files to Windows and Linux servers* <i>* Available for download in the GetIt Package Manager</i>	 
INTRODUCED IN 10.3.3 RIO	Support to deploy Delphi Linux applications in general and RAD Server applications in particular to Docker. <i>* Available for download in the GetIt Package Manager</i>	
WEBBROKER AND DATASNAP MULTI-TIER		
	WebBroker library for HTTP servers development and integration, supporting CGI, ISAPI libraries for Microsoft IIS, and native HTTP servers based on Indy library	   
	WebBroker support for Apache HTTP server modules	   

	WebBroker default encoding changed to UTF-8 rather than ANSI	 	 
	WebBroker deployment support in Linux, as stand-alone or Apache modules		
NEW IN 11.1 ALEXANDRIA	WebBroker deployment support for Android		
	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 OpenSSL client library (now both for REST connections and HTTP connections)		 
	DataSnap server methods support TDBXJSONStream parameter for passing JSON streams between client and server		 
NEW IN 11 ALEXANDRIA	The REST URI is configurable using a new 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), now 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		 
INTRODUCED IN 10.2 RIO	DataSnap REST now explicitly specifies "Content-Type=application/json"		 
ENHANCED IN 10.2 RIO	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		 
	ServerMethods give complete control over functionality of middle-tier. Use TJSONValue pairs making it easier to pass data		 
	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 now 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		 
DATABASE TOOLS			
	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	 	 
	SQL console views for running queries and viewing results on any dbExpress 4 supported database	 	 
INCLUDED DATABASES			
ENHANCED IN 10.3.3 RIO	InterBase 2020 Developer Edition – up to 20 users and 80 logical connections	 	 
ENHANCED IN 10.3.3 RIO	IBLite 2020 for Windows, iOS and Android with unlimited deployment license	 	 
ENHANCED IN 10.3.3 RIO	IBLite 2020 for macOS with unlimited deployment license		
INTRODUCED IN 10.2.3 RIO	Free InterBase ToGo Deployment License for mobile platforms		 
CLOUD SUPPORT			
	Windows Azure components including AzureConnectionString, AzureBlobManagement, AzureQueueManagement, and AzureTableManagement	 	 
	Cloud libraries now use the System.Net native HTTP and HTTPS support	 	 
	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	 	 
ENHANCED IN 10.4 SYDNEY	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			
ENHANCED IN 10.3 RIO	REST Client Library for simplified invocation of REST services, with improvements in MIME types handling, properties (RedirectsWithGET and SecurityProtocols) and events (OnNeedClientCert and OnAuthEvent)	 	 
	Authorization support including Basic Authentication, Plan Authentication, OAuth1, OAuth2	 	 
	TRestClient, TRestRequest, and TRestResponse components	 	 
	REST Debugger tool for testing REST calls and their parameters	 	 
INTRODUCED IN 10.3 RIO	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		

	JSON reader and writers used in the BaaS client library	 	 
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 and mobile platforms	 	 
ENHANCED IN 10.3.1 RIO	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	 	 
ADDITIONAL AND THIRD PARTY INTEGRATED TOOLS			
CONNECT TO REST APIS WITH ENTERPRISE CONNECTORS			
INTRODUCED IN 10.3.3 RIO	Enterprise Connector, powered by CData, allow you to integrate 70+ Enterprise applications, simplifying connectivity into a standard model using SQL*		 
INTRODUCED IN 10.3.3 RIO	Includes components for QuickBooks Desktop, MailChimp, Salesforce, YouTube, SugarCRM, Jira, SurveyMonkey, Amazon DynamoDB, Couchbase, PayPal, eBay, Google Sheets, Facebook, Twitter, Slack, Dropbox and more* *requires separate download from CData web site		 
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			
UPDATED IN 10.4 SYDNEY	Raize Software CodeSite Express integration provides advanced application logging and debugging capabilities* * Available for download in the GetIt Package Manager	   	 
	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	 	 
	Record method calls to add call stack structure to the message log	 	 
IP*WORKS! COMMUNICATION COMPONENTS			
	The full version of /n software's IP*Works! Components – A comprehensive suite of components for Internet communications including more than 40 individual components covering every major Internet Protocol* <i>* Available for download in the GetIt Package Manager</i>	 1  1	 
	Royalty free commercial components for Internet development including ATOM, CalDAV, FileMailer, FTP, HTMLMailer, HTTP, ICMPPort, IMAP, IPDaemon, IPInfo, IPMonitor, IPPort, JSON, LDAP, Mcast, MIME, MX, NetClock, NetCode, NetDial, NNTP, Ping, POP, RCP, REST, Rexec, Rshell, RSS, SMPP, SNPP, SOAP, Syslog, Telnet, TFTP, TraceRoute, UDPPort, WebDav, WebForm, WebUpload, Whois, XMLp, and XMPP	 	 
	Components are easy to use, with a uniform, intuitive, and extensible design; share common interfaces across platforms and technologies	 	 
	Components are fast, robust, and reliable with minimal resource consumption; lightweight and have no dependencies on external libraries	 	 
	Detailed reference documentation, hundreds of sample applications, fully-indexed helps files, and an extensive online knowledge base	 	 
TEECHART CHARTING COMPONENT			
UPDATED IN 10.4 SYDNEY	TeeChart Standard components including chart, DBchart, and a rich set of different business graph types. * <i>* Available for download in the GetIt Package Manager</i>	 1  1	 
FASTREPORT REPORTING SOLUTIONS			
UPDATED IN 10.4 SYDNEY	FastReport VCL RAD Edition reporting tool * <i>* Available for download in the GetIt Package Manager</i>	 1  1	 
	Exports to popular formats (PDF, RTF, HTML, BMP, JPEG, TIFF, GIF, TXT, CSV)	 	 
	Grouping with drill-downs	 	 
	Caching of the big reports(uses less memory)	 	 
	Six levels in master-detail-subdetail relation	 	 
UPDATED IN 10.4 SYDNEY	FastReport FMX RAD Edition reporting tool * <i>* Available for download in the GetIt Package Manager</i>	 1  1	 
SENCHA EXT JS PROFESSIONAL EDITION			
	Ext JS HTML5/JavaScript (Framework and UI Component Library) - Pro Edition		  ARCHITECT ONLY
	Cmd: Build Optimization Tool		  ARCHITECT ONLY
	Ext JS Stencils: Design Kit		  ARCHITECT ONLY
	Architect: Visual App Builder		  ARCHITECT ONLY
	Sencha Themer: Styling Tool		  ARCHITECT ONLY
AQUA DATA STUDIO			
	Register Servers/Databases, Database Navigator/Explorer		  ARCHITECT ONLY
	Query Analyzer (SQL editor) and Query Builder		  ARCHITECT ONLY
	Table Data Editor		  ARCHITECT ONLY
	Visual Analytics		  ARCHITECT ONLY

Schema/Data/File Compare Import/Export Data and DDL	  ARCHITECT ONLY
ER Modeler	  ARCHITECT ONLY
Database Administration	  ARCHITECT ONLY

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

CORPORATE HEADQUARTERS | EMBARCADERO TECHNOLOGIES, INC. | 10801 North Mopac Expressway, Building 1, Suite 100 | Austin, TX, 78759, USA

www.embarcadero.com | sales@embarcadero.com

© 2022 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. 010322