

Borland C++Builder 3 New Feature Benefit matrix December 31, 2024

High-performance C++:

One language, one tool, and one skill-set for creating any Windows application

Unprecedented productivity with a the most advanced, award-winning ANSI/ISO C++ compiler

Support for industry standards	Get full support for industry standards with C++Builder 3, including C++, Win95/NT, Win32 API, multi-threading, ActiveX/ATL, COM, STL, MFC 4.2, OWL, and COFF in order to build the most demanding projects.
Advanced Project Manager makes complex development easier	C++Builder 3 has the new advanced project Manager for the easy compilation on multiple targets such as EXE, LIB, RES, RC and DLL files. Your application may consist of two projects; one for an executable, one for a DLL. Now, you can build both projects at the same time by adding the projects to the same project group. This greatly enhances your ability to share units among multiple projects.
Build Applications Faster	C++Builder 3 gives you drag-and-drop visual development. Over 130 reusable components including menus, dialogs, data visualization flexibility, Win 95 and NT controls, and much more. Build the fastest, most efficient executables for Windows with the world's fastest 32-bit native code compiler.

Add C++Builder's productivity to existing projects with new migration tools

Easy migration of code	C++Builder easily opens your existing MAKE files, converts your rc files from resource workshop, and links in your OWL and MFC code so you can take advantage of C++Builder's productivity more immediately.
New: Resource Converter allows you to use existing C++ code.	<p>This C++Builder release contains a new tool to convert dialog and menu resources from Windows Resource Compiler (RC) scripts into C++Builder forms and units. Placement, size, captions, and style settings of controls in the dialog resources are converted into C++Builder component equivalents with a high degree of accuracy. Now the developer can build upon existing projects with C++Builder's advanced feature sets.</p> <p>The resource import tool also generates matching unit declarations for each converted form, producing a skeleton C++Builder project that is ready to compile and run, saving you from having to manually convert resources.</p>

Advanced Debugging for getting complex projects to market faster

New: Module View for detailed , low-level debugging	Use the module view to see different modules, such as .EXEs and DLLs, within a single debug session. Module view is a three-paned view that shows detailed information about the different modules loaded by the process you are debugging.
New: Event Log for tracking hard to produce problems	You can also display an event log from the debugger. It shows process control messages, breakpoint messages, OutputDebugString messages, and window messages. Using the local menu, you can clear the event log, save the event log to a text file, add a comment to the event log and set options for the event log.
New: Data Watch breakpoints	The debugger now supports data watch breakpoints which help you find complex pointer problems. When set, the debugger breaks if memory gets written to a certain address. You can set a data watch breakpoint two ways:

New: Integrated DLL Debugging	C++Builder 3 allows the developer to debug DLLs within the C++Builder environment. There is no need for a separate and costly debugging program in order to create complex applications containing DLLs. By simply setting the host application and setting a break point in the DLL, the developer saves time in creating and debugging DLLs for use in WebServers and other tools.
New: Automatic Inspection of Local Variables for debugging productivity	While in debug mode, you can show the current function's local variables. To do so, choose Run Inspect local variables and variables in scope will automatically be displayed and updated. This increases debugging productivity.

Active Insight/ATL: An Enterprise Component Foundry for maximum reusability

C++Builder Exclusive: One-Step ActiveX/ATL	C++Builder 3 is the highest-productivity application environment for creating high-speed industry-standard ActiveX components. C++Builder 3 easily creates ActiveXs that can be used with existing development tools, such as Delphi, Visual C++, Java, Visual Basic, or PowerBuilder, in use through out the enterprise. C++Builder 3 ActiveX controls are native machine code compiled for fast performance and do not require a distributable runtime environment.
New: ActiveForms for Internet Applications	Turn any C++Builder form into an ActiveForm. ActiveForms are ActiveX Controls that use the C++Builder form as a container for other C++ components. ActiveForms publish ActiveX property pages and type libraries for adding high-speed functionality to other development environments, for example Internet Explorer, Visual Basic, or PowerBuilder. Or use these forms to deliver applications over the Internet. C++Builder's ActiveForms and Remote Data Broker help deliver ultra-thin, zero-configuration clients in a web delivered multi-tier database architecture.
New: Easy Active Web Deployment	C++Builder 3 uses the ActiveX architecture to easily deploy thin-client, zero-configuration, applications over the web thereby reducing the cost of software configuration and distribution. Web Deployment uses Microsoft standard application delivery mechanisms (File, INF, or CAB File delivery) and the web infrastructures to distribute your applications to a growing number of users on the Internet, Intranet, and throughout the enterprise.
Built in COM Support	C++Builder 3 integrates Microsoft's COM (Common Object Model) seamlessly into its rapid application development environment so that developers can easily and quickly create reusable, language interoperable business objects without having to worry about the complexities of COM. C++Builder uses COM objects in a multi-tier environment to provide re-usable and integrated business solutions.
Integrated Type Library creation reduces the number of programming tasks	The Type Library editor is an essential graphical tool that enables developers to easily examine and create type information for ActiveX controls and COM objects. This type information can be provided with an object either stand-alone in a type library (.TLB) file, or integrated into the EXE or OCX as a resource. Using and creating IDL (Interface Definition Language) has never been easier. Because the TypeLibrary editor is integrated into COM and ActiveX development, the developer has the ability to change things in code or graphically and always have a synchronized view into complex COM development. This will save hours of programming frustration.
New: Efficient Packages creates smaller executables	C++Builder 3 provides a solution for the development of efficient application executables that need to be quickly and easily distributed via an Intranet to a large number of users. C++Builder 3 allows you to recompile with Packages to reduce executable size.

	Packages are a unique leading-edge compiler technology that gives developers the ability to breakdown applications into separate, reusable compiled pieces without a slowdown in performance. Packaged applications use a customizable application environment that is delivered and configured once, thereby reducing both the configuration complexity of the application and reducing the need for additional distribution resources.
--	--

Code Insight: Now, the fastest way to learn Rapid Application Development Client/Server

Code Templates Wizard simplifies coding	C++Builder 3 provides standard code templates for statements such as <i>If</i> , <i>for</i> , <i>while</i> and <i>case</i> so that programmatic syntax is always correct. In the editor, simply start a code statement and C++Builder 3 will complete it for you thereby reducing errors and speeding development. C++Builder 3 also allows the developer to add their own code templates so that the development environment is customized to your corporate standards.
Visual Component Creation for easy code reuse.	C++Builder 3 simplifies the reusability of components by introducing Visual Component Creation. Developers can easily and instantly create combined components with the associated code and place the new component on the C++Builder palette page. Developers can then reuse these objects in the same project or other projects thereby making development easier and faster.

Business Insight: Turn data into actionable information for better decision making

Exclusive: DecisionCube CrossTab helps make decisions faster.	Lend more power to decision support systems by using C++Builder 3's new DecisionCube Crosstab, multi-dimensional analysis of data. Drill Down, Pivot and Aggregation provide new decision support mechanisms for visualizing data and making better decisions without having to buy a full decision support system or write one yourself.
New: Dynamic DecisionGrid	Decision Grid dynamically displays multi-dimensional data so that users see information according to the business need. Pivotable data, collapsable summaries, and a drag and drop interface integrate seamlessly with graphs to drive your business.
New: Powerful TeeCharts makes visualizing data easy	C++Builder 3 provides the fastest way to turn enterprise-wide data into information. Developers use an expert to graphically display enterprise or Internet data with over eleven different styles. For example Line, Bar, Polar, Candle and other chart styles makes interpreting data faster which leads to better and faster decision making.
Enhanced: Integrated Quick Reports	QuickReports lets you design reports visually with native components and the C++Builder form designer. Reports are built with bands and may contain titles, page headers and footers, multiple detail sets, summaries, group headers and footers. Report from any DataSource, including TTable and TQuery or even from arrays, lists or whatever data you want, use the powerful on-screen preview to check your results. Perform calculations like summary and counting of fields automatically. QuickReports also is completely programmable for meeting any reporting need.

Get to the Web: Web Broker for high-speed dissemination of database information over the web:

New: High-Performance Web Server Applications	With C++Builder 3, you create high-speed, high-throughput web-delivered data applications using your existing client/server knowledge. C++Builder's open architecture and support of native system DLLs enables this high-productivity rapid application development for the internet. WebServer applications are server
--	--

	side DLLs that bind directly to ISAPI and NSAPI, web server interfaces from Microsoft and Netscape respectively.
WebBridge an open approach to NSAPI, ISAPI and CGI development	To simplify Internet development, the WebBridge allows developers to program to a common API for both NSAPI and ISAPI. This flexibility protects your code base as the competing Internet standards evolve. The developer can concentrate on implementing business solutions regardless of the back-end web server.
New: WebModules	Web Modules centralize the dispatching of web client requests, the responses for the request, and the creation of HTML content. These web-specific actions combine with data access and business rules to disseminate high through-put data over the web without coding. The WebModule visually controls multiple requests coming into a web sub-site resulting in full client/server functionality over the web using your existing C++ skills.
New: WebDispatcher for more Web Power	The HTTPDispatcher is the component responsible for Web Server Application's event handling. HTTPDispatcher works seamlessly with existing query and table components to produce CGI and HTML applications. This enables the C++Builder developer to produce Web content with the same skill sets used in Client/Server development. Four additional components produce Web content for Queries, Tables, CGI applications and simple HTML

Open Scalable Database Access for high-performance Client/Server applications

High-Performance, Native 32-bit SQL Links with unlimited deployment	C++Builder 3 enhances its' suite of high-performance 32-bit SQL Link native drivers for Oracle, MS SQLServer, InterBase, Sybase, and Informix. It introduces new 32 bit Sybase System 10 and DB2 Native drivers. C++Builder Client/Server Suite includes an unlimited deployment license for all drivers, so it is cost effective.
New: Driver Development Kit for maximum power	C++Builder 3 allows you to create high-speed native driver access to legacy data systems you previously couldn't get to. Native drivers offer greater performance and more power than the lowest common denominator solution, ODBC. The Driver Development Kit (DDK) allows corporations to create native drivers for the Borland Database Engine which complements the new Remote DataSet technology. Native drivers created with the DDK can be used in all Borland's family of products including: IntraBuilder, Delphi, and JBuilder
New: Native Access Database Driver	C++Builder 3 is the only solution for scalability from the Access desktop to the Enterprise. High-speed, native database drivers to Microsoft Access databases combines the easiest to use C++ RAD tool, C++Builder, with the most popular desktop database.
New: Native FoxPro Driver	Enjoy no-compromises support for FoxPro databases within C++Builder 3, the fastest C++ RAD development environment. Developers can now create high-speed C++Builder 3 applications and maintain existing FoxPro databases without incompatibilities. C++Builder 3 has full support for FoxPro indecies which ensures backward compatability for existing applications.
New: Powerful SQL Builder	SQL Builder is a powerful database tool that allows you to create and execute SQL queries without having to remember or know the syntax of SQL. When you use SQL Builder, you can view and edit the resulting SQL. You can create a full range of SQL queries, from the simplest SELECT to a complicated, multi-table join with calculated fields and expression-based joins with point and click ease eliminating errors in queries.

	<p>SQL Builder features include the following:</p> <ul style="list-style-type: none"> • Generates and parses SQL • Supports nested expressions. • Changes row formats dynamically. • Allows users to specify strings without quotes. • Provides robust search and replace capabilities. • Allows users to join and manipulate multiple database tables easily.
Improved SQL Explorer	The SQL Explorer provides a graphical way of managing all your database demands, it supports the creation and modification of tables, indices and aliases. It is an integrated database schema and content management utility tailored to the needs of professional database developers. Comprehensive support for a majority of Oracle, SQL Server, Sybase, Informix, DB2 and InterBase constructs.
Improved Scalable Database Dictionary	The Database Dictionary stores and uses customized information about the contents and behavior of the data in your tables. The data dictionary holds information about extended field attributes like min, max and default values, display preferences and edit masks. Because it is reuseable across forms and applications, you can quickly establish and maintain data integrity. Data Integrity rules from the server or created by the developer can also be stored in the Data Dictionary so that network traffic is reduced when validating server side rules.
Drag and Drop Database Development	C++Builder makes powerful database programming as simple as drag and drop. C++Builder uses the data dictionary to automatically customize the display and edit properties of your data. You select the fields and C++Builder instantly builds all the connections for a live database application saving hours of programming and testing time.
Multiple Database Engine Support for greater connectivity support	C++Builder 3 supports multiple database engines through an open API so that any Database Engine result sets work seamlessly with C++Builder 3's powerful database aware controls. C++Builder 3 uses an open architecture in order to support multiple light-weight database engines. The developer is given complete control to provide the essential database services (e.g. caching, retrieval, language collating, heterogeneous access, generic cursor support) necessary for a specific application. This means that you will never worry about hitting "the wall" with C++Builder.

Advanced Database and Multi-Tier Development Services -- MIDAS Development Kit.

Sets the standard for high-productivity and rapid development of high-performance client, server and multi-tier applications

Exclusive: Remote Data Broker creates thin client database applications for easier deployment and maintenance	C++Builder 3's Remote Data Broker is an essential technology that allows for an ultra-thin client architecture with the lowest configuration and distribution costs. Using OLEnterprise and the Remote Data Broker, the developer can efficiently communicate data from an application server to a client in a multi-tier application environment. C++Builder 3's unique Remote Data Broker reduces network traffic, centralizes transaction management, security, and performance, reduces network loads, and reduces the number of simultaneous database connections.
New: Constraint Broker for automatic constraint propagation. Reduces Network Traffic	C++Builder 3 automatically propagates Relational Database Management Constraints, through the high-speed native SQL Links Drivers, to the client thereby reducing the network traffic associated with validating user input back at the database server. Also propagated from the server are field default values which in combination with constraints help to maintain data integrity through-out the enterprise.

New: Transaction Resolver simplifies multi-tier database development	C++Builder 3's Transaction Resolver gives the developer complete control of a transaction in a multi-tier environment so that they may enforce business processes. The Transaction Resolver allows the developer to respond to transactional conflicts by indicating exactly which piece of information was unsuccessfully updated. This allows the developer to notify the user, change the data, and/or complete the transaction appropriately.
OLEnterprise connects to reusable business objects	OLEnterprise allows for application logic and data access to be centralized in an application server. The application server becomes a one-to-many service provider of business objects and business processes. Therefore business logic can be reused among multiple applications, and because application logic need only be changed in one location maintenance of applications is much easier.
The Business Object Broker keeps applications running 24x7	The Business Broker is a distributed name service that indirectly binds applications to remote objects based on load balancing criteria or object availability. C++Builder 3 uses the Business Broker to ensure that business critical systems remain running in a twenty-four hour by seven days a week environment while ensuring high-speed performance for an unlimited number of users.

Advanced Project Options for flexible projects that run faster, or distribute smaller.

Incremental Linking maximizes productivity	C++Builder 3's incremental linker means those larger projects and subsequent builds go to a running state faster. This means you can write less code between Builds and therefor test code more easily. It also allows the developer to see a running application and check end-user behaviors of that program more often.
Adaptive Compiler Technology delivers faster development cycles	Smart Adaptive Compiler Technology dynamically creates precompiled headers based on developer usage patterns. Now while precompiled header technology has been around since the early eighties, C++Builder adds further optimizations such as memory caching instead of I/O disk caching and a header modification table to reduce the amount of memory needed to store symbol table information. Again, this means less code compiled and faster turn-around times.
Advanced Compiler Options for compatability with your existing code and performance	<p>Not only are there compiler options available for specifically compiling C++Builder projects, there are now options specific to the C++ language. These help reduce the size of code, help with compatibility with existing code and can increase performance.</p> <p><i>Member Pointers:</i></p> <p><i>Compatibility:</i> Support backward compatibility with earlier ANSI versions.</p> <p><i>Virtual Tables:</i> Optimize the size and performance of virtual functions.</p> <p><i>Templates:</i> Enable runtime instantiation of templates classes.</p> <p><i>Exception Handling:</i> Enable/disable exception handling and runtime type information. Disabling exception handling decreases the code size.</p> <p><i>General:</i> Zero length empty base classes</p> <p>Advanced compiler options include:</p> <p><i>Instructions set:</i> Determine the specific architecture targeted by the compilation.</p> <p><i>Data Alignment:</i> Set the block-size starting boundaries for storing data.</p> <p><i>Calling Convention:</i> Determine which calling sequences to generate for function</p>

	<p>calls. The C, Pascal, and Register calling conventions differ in the way each handles stack cleanup, order of parameters,</p> <p>case, and prefix of global identifiers.</p> <p><i>Register Values:</i></p> <p><i>Output:</i></p> <p><i>Floating Point:</i> Optimize performance or precision, and enable emulation of floating point arithmetic with integers.</p> <p><i>Language Compliance</i> Determine which language is used to recognize keywords in your programs.</p> <p><i>Source:</i> Enable/disable nested comments, guarantee MFC compatibility, and set the unique length for identifying variables in your code.</p>
New: Pragma Support gives developers control over the compilation	C++Builder has added new #pragma directives, #pragma option push and #pragma option pop, to enable you to easily modify compiler directives and offer finer control over compilation in general. You can use the #pragma option push to include any number of files that could potentially change many compiler options and warnings, and then, with the single statement, #pragma option pop, return to the previous state.
Intelligent IDE decreases overall link times.	Intelligent IDE automatically determines which Visual Component Libraries need to be used for a given project instead of defaulting to include all libraries. The amount of work the compiler/linker does resolving references is reduced which in turn significantly improves build times.
New: Automatic Version Information makes supporting users easier	Now you can include version information in your Presto projects so that maintenance on delivered applications is easier. In addition to updating version information manually before recompiling a project, you can select the Auto-increment Build Number option. Version information appears on the Version page of the Windows Properties dialog for the project's .EXE file without having to write a single line of code.
Integrated Turbo Assembler (TASM) with MMX support	Turbo Assembler (TASM) is included in Borland C++Builder so that you can optimize time critical portions of your code and use the MMX instruction set. There are over 10 different TASM options that allows the developers complete control over their Assembler code and environment. TASM support means faster performance of your applications in the hands of the end-user.

Support for ANSI Standard C++ helps insure portability

Standard C++ Library from Rogue Wave for maximum power	<p>The Standard C++ Library is a large and comprehensive collection of classes and functions that is Rogue Wave's implementation of the standard C++ language as defined by the International Standards Organization (ISO) and the American National Standards Institute (ANSI). Support for ANSI standards helps insure the integrity of the C++ Language and program portability</p> <p>The ANSI/ISO Standard C++ Library includes the following parts:</p> <ul style="list-style-type: none"> • A large set of data structures and algorithms known as the Standard Template Library (STL). • IOSTream facility • Locale facility • "Templatized" string class • "Templatized" class for representing complex numbers
---	---

	<ul style="list-style-type: none"> • Uniform framework for describing the execution environment, through the use of a template class named <code>numeric_limits</code> and specializations for each fundamental data type • Memory management features • Language support features • Exception handling features • A valarray class optimized for handling numeric arrays
Dynamic Functions for more programmatic control of optimizations	C++Builder offers the programmer more control of the virtual table in order to optimize for speed or size by supporting Dynamic Functions. Dynamic functions are similar to virtual functions except for the way they are stored in the virtual tables. Virtual functions occupy a slot in the virtual table in the object they are defined in, and in the virtual table of every descendant of that object. Dynamic functions occupy a slot in every object that defines them, not in any descendants. That is, dynamic functions are virtual functions stored in sparse virtual tables. If you call a dynamic function, and that function is not defined in your object, the virtual tables of its ancestors are searched until the function is found.
Improved Template Generation Semantics	C++Builder takes advantage of improved Template Generation Semantics which can result in significantly smaller .OBJS, .LIBs and .EXEs, depending on how heavily you use templates. Again, developers are in full control of the important variables that pertain to C++ applications: size and speed.
Exceptional Exception Handling	Write more bullet proof code with ANSI standard Try / Catch constructs or with new support for Try / Finally constructs as they pertain to the Visual Component Library (VCL). In either case, developers can protect memory allocations to help prevent leaks, delegate exceptions for more programmatic control of recovery sections, and instantiate their own exception types for improved error handling.