



What's New in Kylix 3



Borland®
Kylix™ 3
Delphi™ and C++ for Linux®

Borland Software Corporation
100 Enterprise Way, Scotts Valley, CA 95066-3249
www.borland.com

COPYRIGHT © 2002 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. All other marks are the property of their respective owners.

K3-WN-0802

What's new in Kylix 3

Kylix 3 includes enhancements in the following areas:

- “IDE updates” on page 1
- “C++ language highlights” on page 4
- “Delphi language updates” on page 5 (see **Note** below)
- “Web technology updates” on page 5
- “Indy updates” on page 6
- “Database technology updates” on page 6
- “Component library updates” on page 6
- “Delphi runtime library updates” on page 7
- “Open Tools API updates” on page 8
- “Documentation updates” on page 8

Note: The Object Pascal language is now called the Delphi language. The documentation has been updated accordingly.

Not all features are available in all editions of Kylix, as noted in the following sections.

If you are upgrading from a previous version of Kylix, see “Upgrade and compatibility issues” on page 9.

For late-breaking information on installation, registration, upgrade, and compatibility issues, see the PREINSTALL, INSTALL, and README files available at the root of your Kylix installation folder.

IDE updates

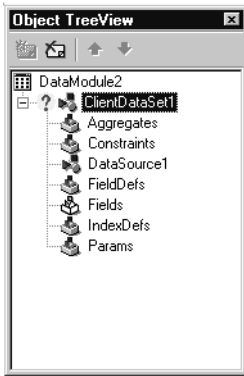
You can now use Kylix to develop cross-platform applications in both the Delphi and C++ languages. Kylix includes an IDE for each language. Use either of the following methods to start either IDE:

- In KDE or Gnome, open the start menu, choose Borland Kylix 3, and select the Delphi or C++ version of Kylix.
- From the bin directory of your Kylix installation, type `./startdelphi` (Delphi IDE) or `./startbcb` (C++ IDE).

Additional IDE changes are described in the following sections.

New Object TreeView

The new Object TreeView displays the logical relationships between visual and nonvisual components on a form, data module, or frame. By default, the Object TreeView is automatically displayed on the left side of the IDE.

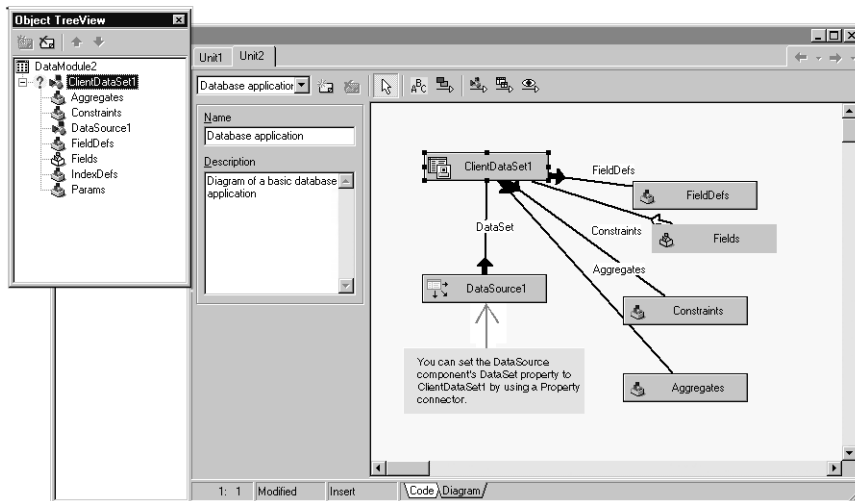


The tree diagram is synchronized with the Object Inspector and the Form Designer so that when you change focus in the Object TreeView, both the Object Inspector and the form change focus.

For more information on the Object TreeView, see the *Quick Start*.

New Diagram page

(Professional & Enterprise editions only) The new Diagram page in the Code editor is a documentation tool for illustrating the relationships among visual and nonvisual components.



You create a diagram by dragging icons from the Object TreeView to the Diagram page. The page displays icons and connecting lines representing the relationships between the components that you place on a form or data module.

For more information on the Diagram page, see the *Quick Start*.

Code Insight changes

- Code completion is now faster and lets you browse to the declaration of items in the code completion list by using Ctrl+click on any identifier in the list.
- New HTML code completion automatically displays valid HTML elements and attributes in the Code editor. (Professional & Enterprise editions only)
- You can create customized code completion managers by using the OpenTools API. See “Extending the IDE” in the online Help for details.

Environment options changes

- The options for the Form Designer on the Tools | Environment Options | Preferences page have been moved to the new Designer page.
- The Tools | Environment Options | Preferences page has a new option, Auto drag docking, that controls whether IDE windows attempt to dock when you move them.
- The Directories dialog box, displayed when editing paths on the Tools | Environment Options | Library page, now displays invalid paths in gray. You can remove invalid paths by clicking the Delete Invalid Paths button.

Component palette changes

- The Component palette has the following new pages:
 - The System page includes several directory and file components.
 - The Indy Intercepts and Indy I/O Handlers pages provide open source Internet protocol components. (Professional and Enterprise editions)
- Right-clicking the Component palette displays the context menu, which includes the new Tabs command for locating tabs on the Component palette.

Debugging changes

- The new Run | Run Until Return command runs a loaded program until execution returns from the current function.
- The new Run | Detach From Program command lets you detach from a process that is currently being debugged.
- The Watch List has multiple tabs, allowing you to organize watches into distinct watch groups for easier debugging. Only the watches on the active tab are evaluated as you debug.

Main menu changes

- The File | New menu command displays five types of projects (Application, Data Module, Form, Frame, and Unit) and Other, which displays the New Items dialog box.
- The new Window command lets you easily locate active windows.
- The View | Component List command lets you multiselect components by pressing the Ctrl key.

Miscellaneous improvements

- From the Project Manager, you can partially compile projects within a group by right-clicking on any project and choosing Make All from Here or Build All from Here.
- The Message view has multiple tabs for displaying different types of messages.
- The Tools | Editor Options | Display page has two new options for displaying tab and space characters in the Code editor.
- The file name tabs on the top of the Code editor can be rearranged by using drag-and-drop.
- Pressing Alt+F10 displays context menus for any selected window, object on a form, or item in the IDE.
- Pressing Alt+Page Down and Alt+Page Up now cycles through tabbed views such as the Code editor, Watch Window, and Message view. These keyboard shortcuts are included in the Default, IDE Classic, and BRIEF key mappings.
- The Project | Export Makefile command exports a makefile that is fully compatible with the GNU Make utility. After exporting the makefile, you can build your project from the command line by typing
make -f ProjectX.mak
where ProjectX.mak is the name you entered for the makefile.
You must source the kylixpath command prior to compiling from the command line, as described in “Compiling from the command line” in the online help.
- The Tools | Build Tools command in the C++ IDE lets you associate file extensions with external programs for execution during a compile.

C++ language highlights

- High-performance 32-bit native-code compiler.
- STLport 4.5 implementation of the C++ Standard Template Library. STLport is a free, open source project that is supported by numerous compilers and platforms. The STLport project uses a BSD-style license; it does not use the GPL. More information on the STLport project can be found at the STLport web site (<http://>

www.stlport.com/). More information on the STLport license can be found at <http://www.stlport.com/doc/license.html>.

Delphi language updates

- Pentium 4 inline assembler instructions are supported.
- The Delphi command-line compiler provides a new switch, `--dynamicloader`, which overrides the default dynamic loader (`/lib/ld-linux.so.2`).

Web technology updates

- The new Web App Debugger enables you to monitor HTTP requests, responses, and response time, and to develop WebBroker and WebSnap applications without installing a commercial Web server.
- WebSnap has a new page producer component, `TXSLPageProducer`, which allows you to use XSLT to build dynamic HTML pages.

Web Services updates

- Web Services applications (both client and server) can now handle attachments. Attachments are sent with SOAP-encoded messages as part of a multipart form. When an application receives the attachment, it saves it to a temporary file, which is then available to your application.
- Exception objects for exceptions that occur when responding to a Web Service request (`ERemotableException` instances) now contain more information from the SOAP fault packet.
- You now have more control over the mapping between invocable interfaces and WSDL documents, including the ability to identify the mapping between function return values and parameter names, the use of namespaces, and default soap actions. On the client side, literal encodings are now supported as well as RPC-style encoding.
- Type definitions are automatically registered with the remotable type registry when you register an invocable interface.
- Invoker components now publish events that let you write code to execute before or after the invoker executes a requested method call.
- `TXSDecimal` has a new `AsBcd` property for easier conversion between XML and native types. Similarly, `TXSHexBinary` has a new `AsByteArray` property. Remotable classes that represent time values now let you work with fractional seconds rather than milliseconds.
- In `DataSnap` applications, the use of `IAppServer` has been changed to `IAppServerSOAP`, which avoids some ambiguities in the `IAppServer` interface. The `UseSOAPAdapter` property of `TSoapConnection` can be used to write clients for servers written with earlier versions of Kylix. `TSoapConnection` also publishes

several new events for you to customize your client applications at various points in the process of executing a Web Service request.

- The IOPConvert interface has a new property: Encoding. This allows you to specify the character set to use for encoded messages that are passed between the client and Web Service provider.

Indy updates

- The Indy components have been updated to Nevrona Indy v9.0.
- The Component palette has two new Internet protocol pages, Indy Intercepts and Indy I/O Handlers. (Professional and Enterprise editions)

Database technology updates

The dbExpress drivers have been updated for Oracle 9i and Informix SE.

Component library updates

- The Component palette has a new System page, which includes several directory and file components.
- The Component palette has two new pages, Indy Intercepts and Indy I/O Handlers, which provide Internet protocols. (Professional and Enterprise editions)
- TOpenDialog and TSaveDialog have been expanded to support additional features such as file previewing.
- The Classes unit includes the following updates:
 - A new exception class, EFileStreamError, has been added. EFCreateError and EFOpenError descend from this class. This new class may take a FileName parameter. As a result, the exception message text now contains the name of the file the error occurred on.
 - The TStrings class has two new properties, ValueFromIndex and NameValueSeparator.
 - The CheckThreadError methods of TThread have been promoted from private to protected visibility.
 - TStream.GetSize has been promoted to virtual protected visibility.
 - The CheckSynchronize function allows you to set a timeout.
 - The new SyncEvent variable provides better thread synchronization support. See the source code comment on this variable for usage information.

Delphi runtime library updates

- The StrUtils unit includes the following changes:
 - There is a new string-searching function called PosEx.
 - Previously, LeftStr, RightStr, and MidStr each had an AnsiString parameter type and return type, and did not support multi-byte character set (MBCS) strings. Each of these functions has been replaced by a pair of overloaded functions, one that takes and returns AnsiString, and one that takes and returns WideString. The new functions correctly handle MBCS strings. This change breaks code that uses these functions to store and retrieve byte values in AnsiStrings. Such code should be updated to use the new byte-level functions described below.
 - New functions LeftBStr, RightBStr, and MidBStr provide the byte-level manipulation previously provided by LeftStr, RightStr, and MidStr.
 - New functions AnsiLeftStr, AnsiRightStr, and AnsiMidStr are the same as the new LeftStr, RightStr, and MidStr functions, except that they are not overloaded with equivalent WideString functions.
 - The new AnsiReverseString function is an MBCS-enabled version of the existing ReverseString function.
- The SysUtils unit includes the following changes:
 - SafeLoadLibrary is now available on Linux (it was previously available only on Windows).
 - The TryFloatToCurr and TryFloatToDateTime functions have been added.
 - The platform-specific constant faSymLink has been added.
 - The new ECodesetConversion exception will be raised when it is not possible to convert between two codesets (between Unicode and a locale codeset, for example, Thai, or vice versa).
- The Variants unit includes the following changes:
 - The VarIsError and VarAsError functions have been added.
 - The EVariantError exception is now a base class for finer grained exception classes that are thrown from variants code.
 - Several new global Variant control variables have been added: NullEqualityRule, NullMagnitudeRule, NullStrictConvert, NullAsStringValue, PackVarCreation, RangeCheckVariants and BooleanToStringRule.
- The Math unit has a new default parameter, RaisePending, in the ClearExceptions procedure.
- The VarCmplx unit has new functions: VarComplexLog2, VarComplexLog10, VarComplexLogN, VarComplexTimesImaginary, and VarComplexTimesReal.
- The StdConvs unit now includes stones in the supported weight units.

Open Tools API updates

The ToolsAPI unit includes the following new features:

- The new IOTAToolsFilterNotifier interface filters output from a build tool and the new IOTAToolsFilter interface registers a tools filter notifier.
- The new IOTACodeInsightServices interface installs customized Code Insight managers and queries registered Code Insight managers.
- The IOTAEditActions interface has new methods, NextBufferView and PreviousBufferView, for cycling through the tabs in the code editor window.
- The IOTAThread interface has new methods, GetOTAXMMRegisters and SetOTAXMMRegisters, for reading and writing to XMM registers.
- The IOTAProcess interface has a new property, OSProcessId, that returns the OS process ID.
- The IOTAMessageServices interface has several new methods that support the new multiple tab functionality in the Message view.
- The IOTAServices interface has new methods to return a number of IDE directories such as the product root directory.
- The IOTABufferOptions interface has new properties, ShowSpace and ShowTab, which control whether space and tab characters are displayed in the code editor.
- A new class, TOTAFile, provides a default implementation for the IOTAFile interface.
- There are two new utility functions, StringToIOTAFile and GetActiveProject.

Documentation updates

- The *Quick Start* and *Developer's Guide* now include information about both the Delphi and C++ programming languages. Icons and headings are used to denote language-specific text and code examples.
- To ensure the continued accuracy of the Kylix tutorials, they have been moved from the *Quick Start* and online help to the Kylix3/documentation/Tutorials folder.

Upgrade and compatibility issues

For late-breaking upgrade and compatibility issues, see the README file in your Kylix installation root directory.

- To upgrade a Delphi (formerly Object Pascal) language project from a previous version of Kylix, open it in the new version. The project is automatically updated to the new release. An informational message is displayed that tells you the project was updated.
- Multi-tier database applications have been changed to use the IAppServerSOAP interface instead of the IAppServer interface. This occurs transparently if you are writing both a new client and a new server. To write a client for a new pre-existing server, set the UseSOAPAdapter property of the TSoapConnection component to false. If you write a server in Delphi, it automatically supports IAppServer as well, to work with pre-existing clients. In C++, servers do not support IAppServer automatically unless you add a call to the RegDefIAppServerInvClass procedure.
- On IOPConvert, the syntax of the InvContextToMsg and MakeFault methods has changed to use a stream rather than a string. This change was required to implement support for the new Encoding property.
- The DispatchSOAP method of IHTTPSsoapDispatch has a new parameter, BindingType, which is required to support attachments. Also, the Invoke method of TSoapPascalInvoker has a similar parameter that serves the same purpose.
- Changes to the StrUtils unit LeftStr, RightStr, and MidStr functions may require you to update code that uses these functions. See “Delphi runtime library updates” on page 7 for details.

