

**Dice™**  
Look to the tech leader first

Get the respect you deserve

FIND TECH JOBS



New to Linux?

web results by **YAHOO!** search[Articles](#) [News](#) [Distributions](#) [HOWTOs](#) [Product Guide](#) [Sponsor Solutions](#) **New**

Search »

[Applications](#) [Servers](#) [Programming](#) [Community](#)[Log in »](#) [Create Account »](#)

## Review: REALBasic 2005 for Linux

Thursday August 04, 2005 (09:01 AM GMT)

By: [Joe Barr](#)

RealSoftware's [REALBasic 2005](#) -- the popular cross-platform interactive development environment (IDE) for Mac OS X and Windows -- is scheduled to debut for the Linux platform later this month, perhaps as early as the LinuxWorld Conference & Expo in San Francisco. For the past couple of weeks, I've been falling asleep each night to the glow of beta versions of REALBasic 2005 running happily on SUSE Pro 9.3 on my monitor.

REALBasic 2005 is nothing like the Basic I used to know. If you haven't used Basic since most jobs required JCL to run, you might not recognize REALBasic as the same language. Sometime since then, for example, the dimension statement has morphed from its assigned role of defining array dimensions to become the defining force for all variables. You might say it has inherited the duties of the DECLARE statement. The language these days is thoroughly modern and object-oriented: it's all objects, properties, methods, and classes.

REALBasic is a powerful language with an equally powerful IDE. It is also a proprietary, closed software product, but you can develop free software with it. That means the source code and the executable of any apps you build with REALBasic can be freely distributed, without restriction. Or if you like, you can build freeware, shareware, or legacy proprietary apps with it. There are no restrictions, no royalties, and no runtime required.

At times REALBasic development can be blazingly, amazingly fast. At other times, however, it can be frustratingly slow and painful, as you struggle to learn the REALBasic way of doing things you might already know how to do a dozen ways in a dozen other languages. It's not that the learning curve for REALBasic is so steep, but rather that at times it seems to go on forever.

You can download the [public beta](#), but remember that it is a beta version; expect bugs and crashes, and expect to submit detailed reports about them.

You'll need a license key to run the beta, but RealSoftware doesn't ask for any personal information before it generates one for you. The company just wants to know how many people are trying the beta. It's one way to measure how large the market is for such a product among Linux users.

### What can you do with it?

If problems in simultaneous application development for Linux and Windows, or Linux and Apple, have held you back in the past, REALBasic just might be the answer to your needs. Ditto if you're currently a REALBasic or Visual Basic developer for Windows and want to extend your grasp to the Linux platform. Cross-platform capabilities is what it's all about. You can configure the IDE to build executables for one, two, or all three environments with just a few mouse clicks.



REALBasic also does networking, with built-in support for TCP and UDP. This is the feature that really caught my eye, as my on-again off-again Internet bridge game project might actually become a reality if the networking chores were easily handled. In fact, that is exactly how I've been exercising REALBasic the past two weeks: by building networked server and client applications to play bridge. Throw in the lure of cross-platform capabilities and the lack of licensing restrictions, and I was ready to give it a whirl.

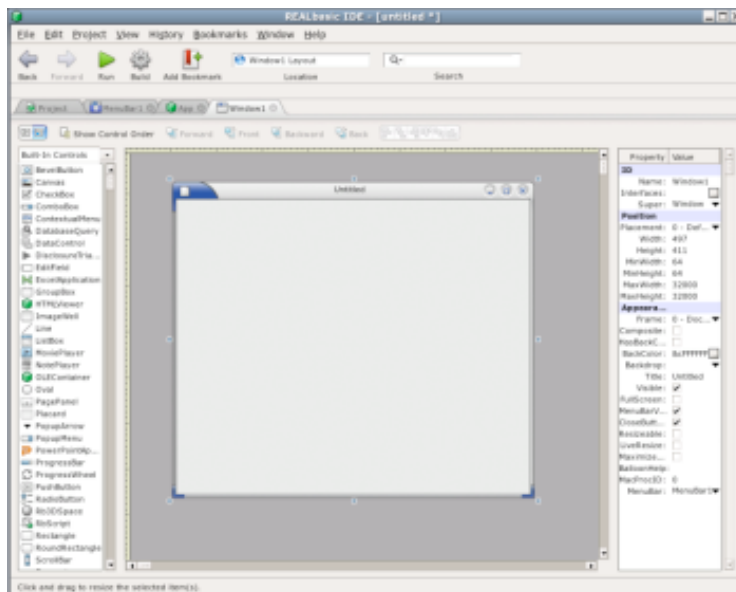
My personal needs aside, REALBasic is probably best for developing small GUI applications for data entry, database queries, or record keeping. But remember, lurking just behind that widget factory that lets you create pretty windows and all sorts of bells and whistles is a powerful language. Games, business, or personal; I don't see any real limits on what it can do.

### A simple example

The image on the right shows REALBasic's object editor screen. You can create a simple standalone application with it in a minute or two. When you start REALBasic, the IDE opens to a Project tab that shows three objects: the application, a window, and a menu bar. To work on the window, double-click on it.

That opens an object editor like the one shown alongside. The window being created is displayed in the center of the screen, REALBasic's built-in controls for windows are displayed in the left margin, and the window's properties are shown on the right. To make the window in our example, you can set the properties for the window title and its background color by clicking on them and either entering text or using a color picker to select the color.

There are four controls visible in the window. I added each by simply double-clicking on it in the built-in controls listing. When a control is added to the window, it has the focus, so the properties displayed on the right side are those for the control instead of the window itself. I added the text for the question and the names of the pushbuttons by selecting the property and entering the text. I positioned the controls in the window by selecting each object and dragging it to where it should be.



Click to enlarge

To add the brains to the beauty, double-click on an object in the window -- the Yes button, for example -- and a code editor replaces the object editor. The left margin then shows the built-in controls available and the methods associated with them. For you old-timers out there, methods are a new way of writing a SUB back in the day.

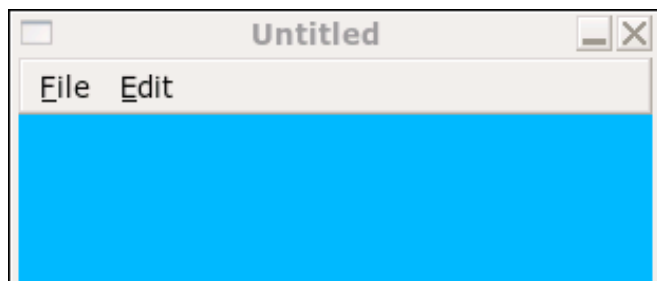
The default method for our pushbutton is the Action method. The code you add here is executed when the button is clicked.

When **both sides of your brain** are satisfied -- the right side with the appearance of the widgets and the left side the logic and syntax of the code -- you can compile and test your application by clicking Run. Once you've cleaned up any errors, you're ready to build standalone executables.

Before doing a build, you need to decide on a name for your masterpiece. By default, it's MyApplication or a derivative thereof, for every possible platform. From the Project Tab, double-click on the app to bring up the application properties in the right margin. Among the properties you'll find the application name on the various platforms. Change them to your liking.

The standard version of REALBasic 2005 for Linux builds a standalone executable only for Linux. If you're running the Pro version, you can change that behavior. Click Project->Build Settings from the toolbar, then put a check mark next to the additional platforms you want: Windows, Mac OS/X and Classic, Mac Classic only, or Mac OS/X only. That's it. Run build, and you've just become a cross-platform developer.

### Not so simple



Real life requirements are never that simple, whether for a personal task like my Internet bridge project, or something to sate your business needs. My project will require two separate apps: a client and a server. Based on the current design, the client will require three windows and at least two TCP sockets. The game server needs only two windows, but at least five sockets. As you can tell by the need for TCP sockets, networking is a big part of the project.

It's a big part of REALBasic, too. It comes with basic and advanced controls for both TCP and UDP sockets. One of the



better known examples of [REALBasic networking prowess](#) on the Net is the creation of a complete Web server with less than a 100 lines of code.

One major distinction between the Standard and Pro versions is the SocketServer class. Its function is to manage all the socket connections that a server might need under the covers, all on a single port. In effect, REALBasic is "dumbing down" the socket code API so that mere mortals can write network servers.

In addition to the SocketServer class, there are two others designed to make networking easy to code: they are the EasyTCPSocket and EasyUDPSocket, both of which are available in both the Standard and Pro versions. Like the ServerSocket, the idea behind them is to make it easier to write network apps.

### The scope is growing

The issue of scope looms larger as you move into the non-trivial programming chores. In REALBasic, you can `declare -- oops`, make that `dim -- variables` in methods where they are needed. But they will only be visible within that method. That's fine for small projects, but not always so fine for larger.

You can get around the scope problem by creating a module -- just click on Add Module -- and then adding properties to it. These properties immediately become visible to all methods in the application. I'm thinking that a module is exactly the right place to put all the housekeeping chores that I need to perform when the application is launched, as well. Adding a method to do that is as simple as clicking Add Method.

As you add properties and methods to a module, you specify the scope for each of them. The default is global. You can require other parts of the application to refer to them explicitly as `ModuleName.PropertyName`, or you can limit their visibility to the Module itself.

The two things that I've scraped my knuckles on the most during my evaluation are the networking between client and server and displaying a hand of cards. Try as I might, I could never get any of the easier socket classes -- ServerSocket or EasyTCPSocket - - to work as I need. Ditto for UDP, which I thought would be ideal for signing on to the server. I've taken a step back and have gotten things working using the plain TCPSocket class instead.

The card display problem is most definitely not due to a shortcoming in REALBasic. It's because I didn't know how to do it in REALBasic, and I never found an example of exactly what I was trying to do that I could copy. Once I learned a suitable technique, it really was simple.

I defined a Canvas control tall enough for a single card image (46 pixels) and wide enough for 13 cards, plus a little extra, to display a hand in.

The plan was simple. The applications deal with a deck of cards represented by their rank in the deck, from 1 to 52. To deal a hand, for example, the server shuffles the deck and then sends each client a message with their 13 cards, separated from each other by a "|".

The client has the cardface image for each card. I carefully created an array of ImageWell objects, and used the editor to load the appropriate cardface image in each. But when I tried to paint the cards, I was told that the ImageWell was incompatible with the DrawPicture and DrawObject methods. That meant I had to do a lot of SELECT CASE routing to end up being able to paint the desired card. Hardly an elegant solution.

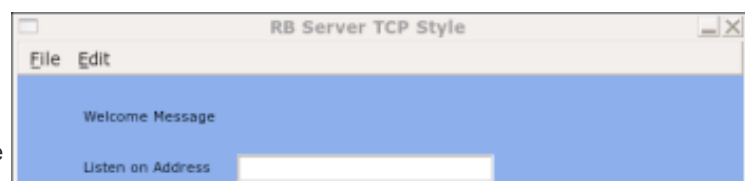
After a couple of days of banging my head on the keyboard, I finally asked for help on the mailing lists. In return, I not only got a lot of help with my specific problem, but learned more about REALBasic as I went. Instead of laboriously loading each image, one at a time, I was shown how to use `GetFolderItem` and load them into an array of type Picture -- not ImageWell -- in 7 lines of code.

Even better, at paint time I could simply reference the array by the rank of the card and paint from it directly. Very nice, even if there was a lot of pain involved in getting there.

I had hoped to have the bridge game finished in time to include it with the story, but I was a little too ambitious for my skills. I'm about 50 percent of the way through the client app, and about a third of the way through the server. The bridge game will have to make its debut at another time.

### Conclusion

There is much more to REALBasic 2005 than I've touched on here, including database functionality. REALBasic supports MySQL, PostgreSQL, Oracle, ODBC, Database4D, OpenBase, REALDatabase, and REALSQLDatabase engines out of the box. Others are supported with plugins. It's also interesting to note that the Linux version of REALBasic 2005 was written with



REALBasic. If that's not eating your own dog food, I don't know what is.

REALBasic is frustrating at times, but I've already accomplished far more with it in two weeks than I ever did with Borland's Kylix. It's also very powerful. If you need a cross-platform IDE that can handle just about any type of functionality you want to throw at it, this might be the right tool for you.

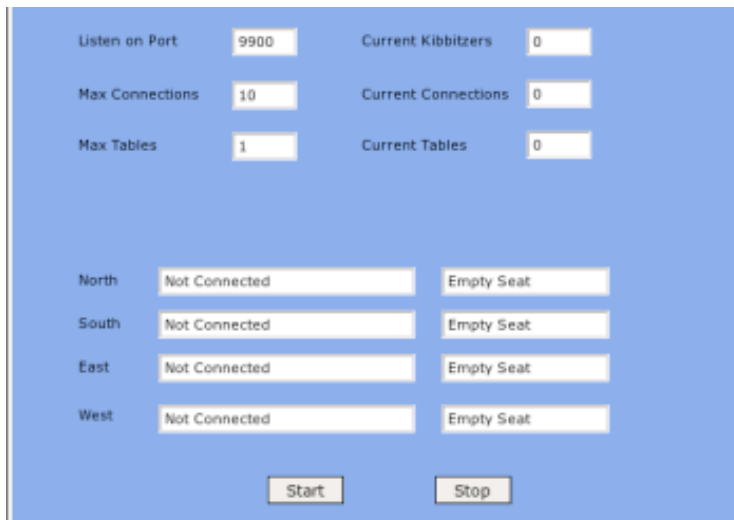
Consider the amount of documentation which accompanies the product, even in its beta stage. The Language Reference manual alone is nearly 1,000 pages. The Users Guide and the Tutorial combined add another 700 pages. And there is a Quick Start manual as well. All are provided in PDF format. The sheer volume of documentation is such that I've learned to love the Search function in Adobe Reader 7.0.

Available support options include a mailing list for the Linux beta. I recommend [signing up](#) for it, especially if you are new to REALBasic or a long time gone from Basic, as I was. Other mailing lists that might be useful include Getting Started, REALBasic NUG, Tips, Games, and Plugins. There is also at least one unofficial IRC channel: #realbasic on FreeNode.

Will this be the first proprietary IDE to really catch on with Linux development? The REALBasic customer base began in the Apple community. But since a Windows version began shipping in 2003, Windows users have been a larger and larger share of that base. Currently, the ratio is roughly 55 percent Apple, 45 percent Windows. Previous releases have been able to build applications for Linux, but you couldn't develop them on Linux.

The Standard Version of REALBasic 2005 for Windows and Apple costs \$100, but for Linux it is free. Surprisingly, that free-as-in-beer price tag will remain even after the beta is over. The Pro version, which offers additional built-in functionality, including the ability to compile for platforms other than the platform it is running on, currently sells for \$400, and that's what the Linux version will cost as well.

REALBasic allows me to produce GPL-licensed applications, and to do it with a free-as-in-beer development tool that is far more capable than my meager skills can take advantage of. And if I want my friends who use Macs or Windows to be able to play bridge with me, REALBasic 2005 for Linux is likely to become the first application I've purchased for my Linux desktop in a long time.



## Related Links

- [public beta](#)
- [both sides of your brain](#)
- [REALBasic networking prowess](#)
- [signing up](#)
- [REALBasic 2005](#)
- [Best deals: Programming](#)
- [More Programming stories](#)

[< Review: HP Deskjet 6840 on Linux](#)

## Comments on [Review: REALBasic 2005 for Linux](#)

[Post to discussion](#) - [Log in/Create an Account](#) - [Return to top level](#) - [1 comments](#) - [Search Discussion](#)

Display Options:

**The Fine Print:** The following comments are owned by whoever posted them. We are not responsible for them in any way.

**The review would be more useful...** (Neutral)

By [Prototerm \(196579\)](#) on 2005.08.04 9:16 ([#85197](#))

...if the author had been able to compare RealBasic to MS Visual Basic. Most programmers writing Basic programs these

days use VB on some version of Windows. Comparing the product to programming in C, C++, or Pascal (that's what Kylix uses) isn't very informative. Yeah, we know Basic is different from C. So what? If you're seriously looking at this product, chances are you're already programming in Basic, so the choice of language is already made.

Some issues that would be helpful to compare:

- \* The components available in the VB toolbox.
- \* Third party components on multiple platforms (and how cross-platform are they?)
- \* Is there any cross-platform support for things like automation (COM and DCOM). Yeah, it's a Windows thing, but it would be nice to know if they worked out some sort of equivalent for non-Windows platforms, using Interprocess Communication?
- \* Is there an equivalent to database connectivity components such as ADO, and what databases are supported.

Well, you get the idea.

[ [Reply to This](#) ]

---

## OSTG Services

---

[Linux.com Product Guide](#) - Read Reviews on Hundreds of IT Products

[OSTG TechJobs](#) - Search Thousands of IT Jobs Now

[OSTG Broadband](#) - Lightning-fast Connections for Power Users

[OSTG Whitepapers](#)

### Special Offers

Get special offers on:

Linux  Application Dev  Programming  Software    Email:

### FREE WHITEPAPERS

[Detect and Protect Your PDAs and Smartphones](#)  
[Enterprise Secure Instant Messaging White Papers](#)  
[Get a complimentary report on EAS market vendors](#)  
[Competitive Token Upgrade](#)  
[Enterprise Secure Instant Messaging White Papers](#)  
[Anatomy of a Native XML Database](#)  
[Avoid TimeSheet Software Failure](#)  
[ITIL Explained](#)

[More WhitePapers](#)

---

### Ads by Google

#### Java Installer

install4j creates installers and launchers for all major platforms.

[www.install4j.com](http://www.install4j.com)

#### Integrated Development

Embedded Development Tools Try CodeWarrior Tools Today!

[CodeWarrior.com](http://CodeWarrior.com)

#### Source Code

Free Source Code info from the experts at the Tech Encyclopedia.

[www.Tech-Encyclopedia.com](http://www.Tech-Encyclopedia.com)

---

---

Sponsor Links

---

[PERVASIVE POSTGRES. NO fees. YES certified.](#)  
[FREE Object-Oriented DBMS Download](#)  
[64-bit-enabled IBM eServer\(tm\) xSeries\(r\) Systems.](#)  
[Dedicated Servers](#)  
[Interland: Dedicated Hosting](#)  
[OSDL Enterprise Linux Summit: Come To the Source](#)  
[FREE trial of VMware Workstation 4 click here!](#)  
[Secure, Dependable PC Platform from ATI](#)

---

-advertisement-

© Copyright 2005 - OSTG, Inc., All Rights Reserved  
[About Linux.com](#) - [Privacy Statement](#) - [Terms of Use](#) - [Advertise](#) - [Contact Us](#)  
[NewsForge](#) --online technology news on Open Source, Linux and IT.  
[RSS](#) Add our feed to your site