



# Creating TrueApps *and using the* WebBASIC Reader

## Creating TrueApps™

The True BASIC Programming Language has been designed to allow you to write your source code once and have it run with a minimum of change on many different operating systems. TrueApps™ make this possible. TrueApps are slightly modified *compiled* True BASIC files that you have learned to create with this BRONZE Edition.

When you first write a program in True BASIC, you are working with source code. This textual form of the program is easy for authors to create and manipulate. It contains lines such as:

```
LET i = 10
```

True BASIC source code is the same for every operating system and every function and statement is supported in every version of the Language System.

When you are satisfied that the program you have written is complete, you go to the **Run menu** of your True BASIC authoring package, select the **Compile** function and create a compiled version of your program. This step translates the source code into a form that not only compresses the code but also allows it to be executed more rapidly. The output of this step is often called byte code and is not easily readable by humans. A **TrueApp** (formerly known as a compiled True BASIC program) is now not only 'lean and ready-to-run', but also portable and a great way to exchange programs on the Internet.

## WebBASIC Readers

To exploit the possibilities of portable, mini-applications, a Reader for each operating system was needed. The **WebBASIC Readers** now perform that task, and best of all they are free and always available for instant downloading from the True BASIC website (<http://truebasic.com>). WebBASIC Readers are available for MS-DOS, MacOS, Windows 3.X, Windows95, and WindowsNT. Readers for other operating systems will be added as demand is demonstrated. The average size of a WebBASIC Reader is 800K.

The heart of a **WebBASIC Reader** is a portable set of the run-time environmental features that are added to a True BASIC compiled file in the **bind** step used to create independent free-standing applications.

This allows you to write a **TrueApp** on your favorite operating system and have it run, without change, on all other operating systems that are supported. This is exactly the functionality that Java is trying to re-create for the C++ language. You already have it in your favorite programming language, True BASIC.

## Suggestions for Writing TrueApps

Writing programs as TrueApps to be used with the WebBASIC Readers is quite easy if you remember these two issues: the location of supporting library files, and what (if any) additional resources your program requires to function.

If the user simply drags-and-drops the TrueApp onto the WebBASIC Reader, the Reader gets the disk location for that file. This makes it easy to predict the location of the library files relative to the TrueApp.

If the WebBASIC Reader is run as a 'helper' application from a web browser (such as Netscape or Internet

Explorer), the browser will download the a TrueApp to a temporary location and then pass that pathname as the argument to the Reader. This, however, can be more of a problem to predict, and worse yet, it often varies by platform and browser.

Here are ways you can cope with this:

- [1] Use absolute path names in the LIBRARY statements in the program.
- [2] Concatenate the compiled library file onto the end of the compiled main program. This will increase the size of the TrueApp, but ensure that all the libraries will be available to the TrueApp.
- [3] Install any required libraries in a location which is constant relative to the location of your web browser.

If you chose the second option and are concerned about the size of the TrueApp file, you may wish to write a routine which parses the library file and includes only those parts which are used by the main program or other library subroutines or functions.

Once you've created your TrueApp on a Mac OS machine, you'll need to correctly set the type and creator codes to use them locally. We've included a handy DO program called **trcTOapp.trc** which accomplishes this. To use it, just compile and save your program, then choose **>AppMaker** from the **Run** menu. When prompted, select the compiled program whose type you want to change and click OK.

## Web Browser Configurations

When a web server delivers a page to a client, it prefaces that page with a descriptive header. The header uses a classification system called MIME types to describe the content of the page. MIME stands for **M**ulti-**p**urpose **I**nternet **M**ail **E**xtensions. (You can learn more about this RFC1521 standard at <http://www.isi.edu/rfc-editor> and <http://www.internic.net/ds/dspg1intdoc.html>). In our case, we want to instruct our browser to send pages with MIME type **application/vnd.trueapp** to the WebBASIC Reader.

Using this standard, *application* is called the **type** and denotes that the content is a program. The second part, *x-truea* says that this is a vendor-specific **subtype** called *truapp*.

A **server** that is correctly configured will send a header line that says "Content-type: application/vnd.trueapp" (and probably several others) before sending the TrueApp on its way. This will tell a correctly-configured browser to run the file using the WebBASIC Reader.

Configuring a **client** to use the WebBASIC Reader is also quite straightforward.

When a web server delivers a page to a client, it prefaces the content of the page with a header. This helps determine how the content is to be used: display it in the browser, save it to disk, display as text, or run a plug-in or helper application to use it. We need to configure our browser to do the latter.

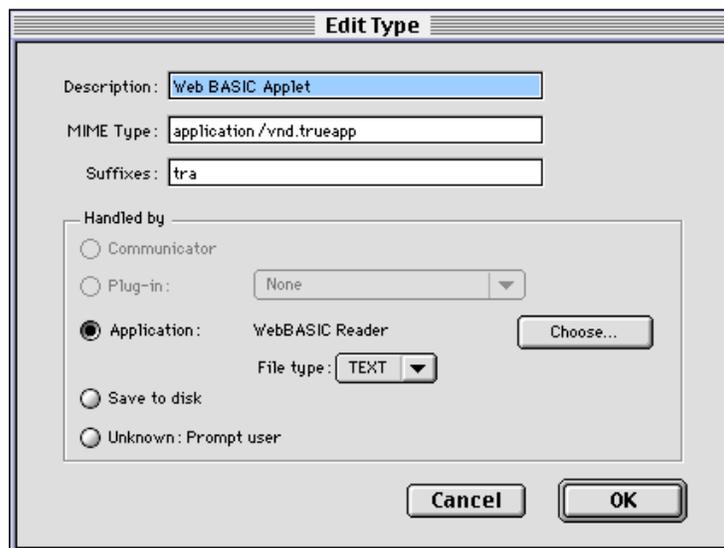
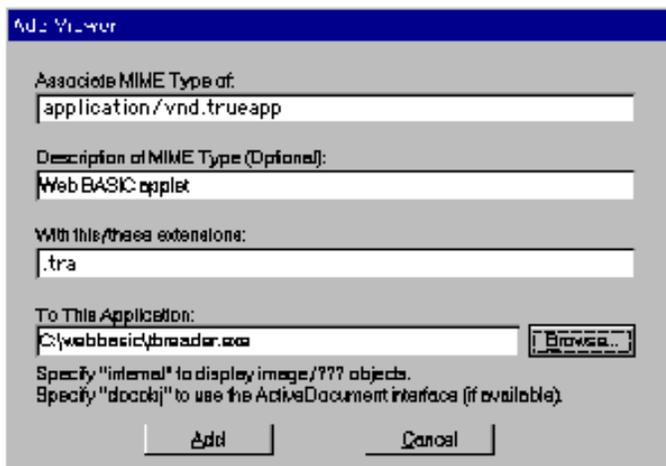


Figure F.1 – Netscape Browser

This is the setup procedure for Netscape Navigator 4.x for MacOS:

- [1] Choose Edit from the menu.
- [2] Choose Preferences.
- [3] Choose Navigator, then Applications in the right-hand pane.
- [4] Click the New button in the left-hand pane.
- [5] Enter “WebBASIC Applet” in the **Description** field
- [6] Enter “application/vnd.trueapp” in the **MIME Type** field
- [7] Enter “tra” in the **Suffixes** field.
- [8] In the “Handled by” box, click the “application” radio button; click the “choose” button and browse to select the WebBASIC reader application icon.
- [9] Click **OK**.

This is automatically set up by the installer for Windows browsers (either *Internet Explorer* or *Netscape Communicator & Navigator*)



*NCSA Mosaic 3.x for Windows*

If these settings are not in place before you try to run TrueApps, your browser will be filled with garbled text as the TrueApp is downloaded, and the browser you are using will have no clue about the application required to run this compiled byte code.

You can obtain Personal Web Servers from the Microsoft website. It is available for MacOS and several Windows platforms. It is downloadable as part of the Internet Explorer 4 package.

## **Learn More About TrueApps at our Web Site**

We've shown you the primary steps you need to take to deliver TrueApps that will automatically launch the WebBASIC Reader when downloaded.

If you want to experiment with some already created TrueApps, you can go to our website at <http://www.truebasic.com/trueapps> to check out the latest additions to a growing library of solutions. Each TrueApp carries a short description telling you the functionality of that particular program.

We invite you to submit examples that you have created. It is easy to take existing True BASIC source code, add the appropriate libraries and compile it into a TrueApp. Submit it as an e-mail message attachment to [trueapps@truebasic.com](mailto:trueapps@truebasic.com) and, with your permission, we will post it at our website.

This new approach will enable you to include computer programs as part of a book, research article, or other report which would benefit from interactivity. Licenses are available that will enable you to include the various WebBASIC Readers with items you wish to distribute, either in software form or across the Internet.

Another beauty of the TrueApps approach is that you can distribute program solutions to accomplish several objectives.

- [1] **Open approach:** Include both a True BASIC source code file *and* a compiled TrueApp file. This allows other users to run your program and also see how you wrote the original program. This also allows them to modify or extend your code as they perform their own experiments.
- [2] **Intellectual Property Approach:** Providing only a TrueApp file allows a user to run your program but does not allow them to see your original source code.
- [3] **Commercial Approach:** And, of course, you can always take your compiled True BASIC file and bind it as a free-standing application. To do this, however, you need to purchase a True BASIC Language System for each platform you support and bind your code for each operating system.

You should be able to compile and post your first TrueApps in a very short time, if you pay attention to the following areas:

- [1] Be sure to include the supporting libraries with the TrueApps you create.
- [2] Document the functions of your TrueApp with a ReadMe file or as program comment lines if you are also providing a source code file.
- [3] Join the True BASIC Maillist. Send a message to: `tbforum-subscribe@truebasic.com`. This is a great way for you to ask questions and compare notes with other persons preparing or using TrueApps and the WebBASIC Readers.

**TrueApps™** and the **WebBASIC Readers™** were created in response to requests for a no-cost or low-cost way to distribute programs that would run (without re-writing) on a variety of operating systems. The possibilities are limited only by your imagination!