

Simple Synergy—ZipQuest Pro Meets/and/& System7 Pack

Theme

Automated ZIP code entry seems to be one of those “little” features. It doesn’t have the flash and panache of multiple processes nor does it have AreaList’s drag and drop capability. However, it is a feature that enhances the ease of use and improves the data accuracy of your systems.

In addition to decreasing the work of data entry, automated ZIP code lookup decreases the chances of error during data entry. Thanks to accurate ZIP-coding, the chance of sending a products or invoices to the wrong address is reduced. In addition, the value of marketing analyses increase.

In short, accurate ZIP coding is a winning situation for all involved.

General Background

Developers have been able to provide ZIP code lookup in 4D for a number of years. One approach it to find a text file that contains ZIP code information and store the data into records in 4D. While the *price* of the text file is minimal, since it is available from online services, the *cost* of the 4D-only approach may be significant. In addition to the need to download a text file every few months, the disk space and maintenance requirements for hundreds of thousands of indexed fields makes this a less than attractive approach.

A convenient alternative that this author has benefited from over the years is to take advantage of the products created by Montage Software.

Montage Software

The products from Montage Software perform high speed searches in their information-packed data file. Over the years, Montage has greatly increased both the features of their software and the information available from the datafile.

GetZIP

The GetZIP external was created in August of 1988 and it consisted of an external and a text datafile. To use GetZIP, the User entered the ZIP code and the external would return the

City and State. This product is very quick and is still in use in some systems today.

GetZIP was a freeware product which met great acceptance in the 4D community.

ZIPQuest and ZIP Quest Plus

To improve on GetZIP theme, ZIPQuest and ZIPQuest Plus were released in 1993 as one commercial package. ZIPQuest provided the City, County, and State information, whereas the more capable ZIPQuestPlus also returned the Federal Information Processing Number, Area Code, and TimeZone.

ZipQuest Pro

In August, 1994 Montage released ZipQuest Pro which has proven to be an extremely versatile product. ZipQuest Pro which provides more data than its predecessors as well as improved ways to retrieve that data.

The new version includes both an external and a Macintosh application. In addition to performing a ZIP code lookup, a major enhancement to the external allows you to perform a "distance lookup". When the external is passed two valid ZIP codes, it will return the straight line distance between them. When the standalone application is run, it provides not only the ZIP code lookup and the distance calculation, but it allows you to do a wildcard lookup of city names. The other advantage to ZipQuest Pro is that it is Apple Event savvy. With this feature, an Apple Event savvy *developer* can run ZipQuest Pro on one computer and query that copy of ZipQuest Pro from any client machine running a 4D database.

This last capability can provide significant "bang for the buck" with your clients!

To take advantage of the Apple Event capability, the author used System7 Pack from ISIS International. System7 Pack allows tremendous flexibility, has with a "no hassle" license, and has *tremendous* technical support.

The combination of ZipQuest Pro and System 7 Pack creates an inexpensive ZIP code lookup system with excellent response time, a small datafile, yet it requires only a few lines of code.

The Nitty Gritty

Setup the Server

ZipQuest Pro has requires minimal hardware. In fact, it needs only a Mac Plus or later, running System 7.0.1 or later.

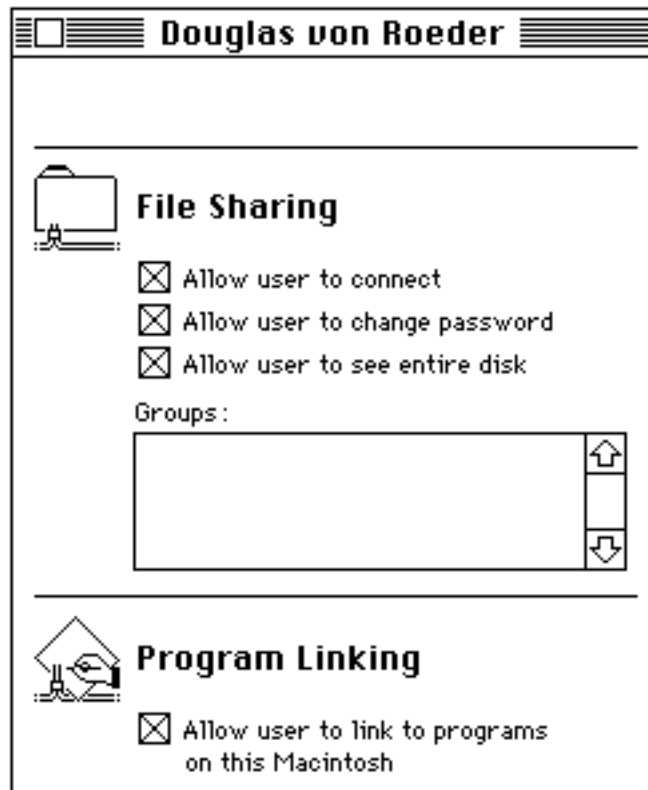
ZipQuest Pro takes up 1.4 megabytes of free disk space and requires only 120k of RAM.

If you have a workstation dedicated to batch processing, ZipQuest Pro can be launched there or it can be run on the same Macintosh as 4D Server. Under test conditions, ZipQuest Pro had no perceptible impact on server performance.

Installation is very simple—drag ZipQuest Pro and its' datafile into a folder, enable Program Linking for ZipQuest Pro, and double click.

To enable program linking for ZipQuest Pro, click on the ZipQuest Pro application icon in the Finder and select the Sharing menu item from the File menu. When your Macintosh opens the Sharing window, check the Program Linking checkbox.

To allow Users to communicate with ZipQuest Pro, you will have to create a new User in the Users and Groups Control Panel. You will have to make sure that your enable Program Linking for each User!



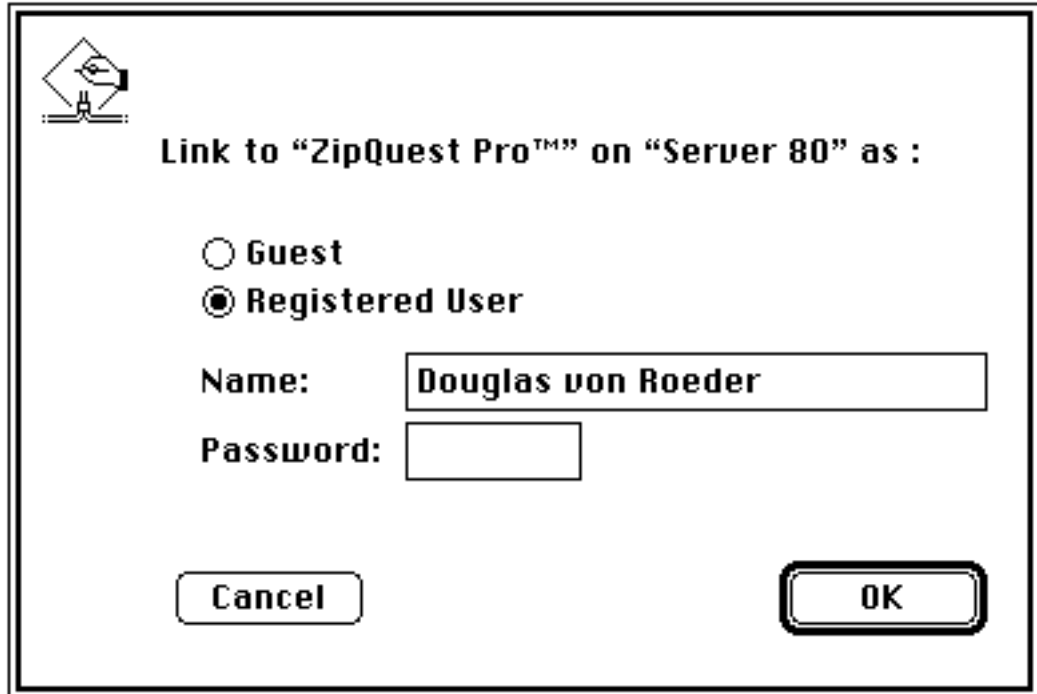
The 4D Client end requires code for data entry and Apple Event communication with ZQP.

Setup the Client

There are two segments of code that need to be programmed in the 4th Dimension application. The modules to be programmed are modules to allow the ZIP code to be data entered and, finally, a last segment of code to handle the query and response duties with the ZipQuest Pro “server”.

The Apple Event Code

Since Apple Events and Program Linking will be used to connect to ZipQuest Pro, the ZipQuest Pro server will try to open the Program Linking dialog box, shown here.



This dialog box may be annoying to the User since it will open the first time a User tries to establish a connection to ZipQuest Pro. ISIS International provides an external called “AutoGuestPatch” which solves this problem very easily. AutoGuestPatch can be called in the Startup procedure and it will allow a transparent connection to ZipQuest Pro. The one line of code needed to do this is shown here:

AutoGuestPatch (1)

The Data Entry Code

The script on the ZIP code field or variable calls a procedure named “Get_ZIP”. The code for the script is shown here:

```
Get_ZIP (Self»;»[People]City;»[People]County;»[People]State;»[People]Phone;»[Peopl
```

Making the Connection

The procedure “Get_ZIP”, shown below, is responsible for passing the query to ZipQuest Pro, receiving the message

that is returned, parsing the message, and assigning the data.

The local string variables \$ServerName and \$NetName are used to determine the network address of ZipQuest Pro. In the sample code, the Chooser name of the Macintosh that runs ZipQuest Pro is "Server 80". In addition, since the network has only one zone, \$NetName is assigned a value of "*" .

Once the address has been determined, two Apple Events are sent. The first Apple Event is an instruction to ZipQuest Pro to perform a lookup of the value passed as \$1. This is done by passing the "LKUP" command.

The next step requires the *SendWithReply* command which instructs ZipQuest Pro to send the results of the last search in a specific format.

ZipQuest Pro uses a simple but effective syntax to specify how the data are formatted. The table below, taken from the ZipQuest Pro manual, details the formatting capability.

Token	Associated Data Element
ZI	ZIP Code
Ci	City name with first letter in uppercase and the rest in lower case
CI	City name in uppercase
ST	State (abbreviation only)
Co	County name with first letter in uppercase and the rest in lowercase
CO	County name in uppercase
FI	FIPS#
AC	Area Code
LA	Latitude
LO	Longitude
TN	Time Zone by name
TZ	Time Zone by number
\T	Tab
\R	Carriage return

The data are returned in the variable vtZIPReply. If the contents of the variable passes a simple error check, the vtZIPReply is parsed and the data are assigned to their respective fields.

The "Get_ZIP" procedure is shown here...

```

Case of
: (␣vonR)
  `© 1994 SE Software
  `Procedure: Get_ZIP
  `August 28, 1994
  `Douglas von Roeder

```

` \$1 = the ZIP code
 ` \$2 = a pointer to the City field/variable
 ` \$3 = a pointer to the County field/variable
 ` \$4 = a pointer to the State field/variable
 ` \$5 = a pointer to the field/variable that will receive the Area Code
 ` \$6 = a pointer to the TimeZone Name field/variable

`... \$ServerName and \$NetName MUST be changed for each installation

End case

C_TEXT(vtZIPReply)

C_STRING(5;\$1)

C_STRING(35;\$ServerName;\$NetName)

C_POINTER(\$2)

C_LONGINT(\$Target;\$i)

\$ServerName:="Server 80"

\$NetName:="" `1 "*" indicates that there is only 1 zone

`1 Detm the network address

Err:=**StringToAddr** (\$ServerName+":PPCToolBox@"+\$NetName;"ZipQuest Pro™";"ZQ94e

`1 Send the query to ZQP

Err:=**AESend** (\$Target;"ZQ94";"LKUP";\$1)

`1 Send command to return the formatted response

Err:=**SendWithReply** (\$Target;"ZQ94";"FRMT";"Ci\RCo\RST\RAC\RTN";vtZIPReply)

If (\$Err#0)

ALERT(\$1+" is not a valid ZIP code.")

Else

For (\$i;2;Count parameters)

 \${\$i}»:=**Get_NextItem** (»vtZIPReply;Char(13)) `1 Parse the ZQP reply

End for

End if

This combination of System7 Pack and ZipQuest Pro allow you to create a high performance, low cost ZIP code lookup system. By providing this feature for your clients, you will reduce the data entry effort, increase the accuracy of data entry, and improve the deliverability of their mailed correspondence. This looks like a "win-win-win" opportunity.