Cool Tool of the Month

by Whil Hentzen

One of the joys of working with FoxPro is the richness of the environment. Because of the accessibility via the API and the fullness of the language, a flood of third party products is available. One of the most common questions I get asked is "What utilities and libraries do you use?" As an independent developer and through my work with The Ultimate FoxPro Reference, many of these third party tools cross my desk. The purpose of this column is to share with you the best, the most interesting, and the most useful of these tools.
We're starting this column with a shareware utility that belongs in every developer's toolbox. Joe Gotthelf's jKey provides the ability to do incremental searches while in Browse.

The Problem and The Solution

Many developers present a selection list to their users via a Browse, but the Browse interface requires the user to use the cursor or Page Up/Down keys to navigate through the entries. In all but the most trivial cases, this process ranges from inconvenient through annoying and all the way to unusable. jKey provides the ability to incrementally search during this Browse.
An incremental search allows you to select an item in a list by typing the first few characters of the entry. For example, after you call up FoxHelp, typing "D" moves the cursor to the first entry in Help beginning with "D" - Date. Successive letters, if typed rapidly enough, move the cursor to that entry - for example, typing "DEL" moves the cursor to DATE, DEACTIVATE, and finally DELETE. Thus, by using jKey, you could present an alphabetized list of customers in a Browse, and your user could select a customer just by typing the first few characters of the name.

How To Use It

jKey is a function written in C that is installed in memory via the SET LIBRARY TO command. After it is resident, it can be called like any native FoxPro function. It will intercept the keystrokes pressed and perform a SEEK on the expression. Here are the basic steps:
1. After jKey is resident, select the database to be searched and SET ORDER TO the index tag that will be used for searching.
2. Call the jKeyInit function. jKey will now intercept all keystrokes and perform a SEEK.
3. Issue an ON KEY statement to trap when the user selects an entry (by pressing ENTER, for example). This step is not required but is pretty handy.
4. Issue a BROWSE command. Since you are actually Browsing a database, you might want to qualify the fields to be Browsed and perhaps specify them as NOEDIT.
5. Once the user has left the Browse (say, by selecting your ON KEY hotkey), issue the jKeyCanc command in order to disable jKey (otherwise, successive keystrokes will continue to be processed by jKey) and, if appropriate, clear out the ON KEY.
6. Remove jKey from memory (if desired).

jKey has a number of options and features that I've not described here. The documentation supplied with jKey explains what they are and how to use them, and includes several sample programs.

Notes

There are separate versions of jKey for FoxPro 2.0, 2.5/DOS and 2.5/Windows. I've used jKey in a variety of applications under 2.0, 2.5D and 2.5W on several different machines. Some people have run into "bnormal program termination" errors with 2.5D - mine were due to an overly aggressive install of QEMM's Stealth on a Compaq - treading a bit lighter solved everything.

A UDF for jKey

Using jKey is not difficult, but it still requires a half-dozen lines of code, and more if you're going to do different things with it. I use jKey a lot, so I wrote a short function that I call. The parms passed control the number of columns in the browse and the column headers. The size of the browse window is automatically calculated from the size of the fields.

```
*        Name: ZJBROW.PRG
*      Author: Whil Hentzen
* Description: passes parms to jKey
*            : 1st 4 parms just call jKey
*            : last 2 parms add a third field to the Browse
*            :
*      Update: FoxPro 2.5/DOS
*            :
*   Called By: common library function
*            :
*      Passed: cDBF, cIDField, cNameField, cTag, cJKeyPrompt
*            : cExtraField, cExtraHeading
*           :
*   Coming In: cDBF is already open and in current WA
*           :
* Files Req'd: jkey.plb
*           :
*      Impact: positions record pointer on record chosen
*           :
*     Syntax : m.cID = zjbrow(dbf, id field, name field, ;
*           :                tag, jkey prompt)
*           :   OR
*           : m.cID=zjbrow("product", "cIDProd", "cNameProd", ;
*           :              "ucIDProd", "Type Product Name: ")

para cDBF, cIDField, cNameField, cTag, cJKeyPrompt, cExtraField, cExtraHeading

* set up database and jKey
set orde to (cTag)
go top
wait clea
set libr to jkey
set talk on
on key labe ENTER keyboard "{CTRL-W}"
=jkeyinit("U","", cJKeyPrompt)
set talk off

* define a window wide enough to accomodate all the fields
* display the Browse
* after the user is done, deactivate jKey and the Browse
do case
case para() = 7 && extra field/title passed
  defi wind wjKeyBrow from 4,1 to 18, ;
  fsize(cIDField) + ;
  fsize(cNameField) + ;
  fsize(cExtraField) + 6
  acti wind wjKeyBrow
  brow noedit fiel ;
   (cIDField) :H="ID":W=.F., ;
   (cNameField) :H="Name", ;
   (cExtraField) :H=cExtraHeading ;
   in wind wjKeyBrow
  =jkeycanc()
  rele wind wjKeyBrow
case para() = 5
  defi wind wjKeyBrow from 4,1 to 18, ;
    fsize(cIDField) + ;
    fsize(cNameField) + 6
  acti wind wjKeyBrow
  brow noedit fiel ;
   (cIDField) :H="ID":W=.F., ;
   (cNameField) :H="Name" ;
   in wind wjKeyBrow
  =jkeycanc()
  rele wind wjKeyBrow
endc

on key labe ENTER

cID = eval(cIDField)
return cID
```

Where To Find It

The jKey package (a self-extracting file named JKEY25.EXE) consists of about a dozen files - the three versions, documentation, sample programs, and a couple of files of particular interest to international users. Now that you're convinced that jKey is one of those "gotta haves", remember that this is shareware - if you like it and use it, send the author $30. You can find it on this month's source code diskette or in CompuServe's FoxForum libraries. You can also order it directly from Joe at August Technologies, 41 Willowbrook Rd, Broomall PA 19008-1749 for $30.

Whil Hentzen is president of Hentzenwerke, a 10 year old computer consulting firm that specializes in FoxPro and TRO-based strategic database applications and has commercial products and custom applications running throughout the U.S. and in 12 foreign countries. Author of The Ultimate FoxPro Reference, Whil heads up the Milwaukee Association of FoxPro Developers and has spoken on Rapid Application Development at developer conferences and user groups across North America. He can be reached at 414.332.9876 (voice), 414.963.4999 (fax), 70651,2270 (CompuServe).