Editor Archaeology

Article
Robert Cemper · Jul 4, 2020 4m read

Editor Archaeology

During the development of the Terminal Multi-Line Command Editor I discovered in my IRIS installation a piece of software that I just can classify as a historic artifact. And it is still fully operational !!!

As it dates back to times before InterSystems was founded in 1978 you may understand my surprise.
I personally stepped into that environment in 1978 and used it then for daily work.

First some short explanation or the environment:
The machine had 128 kBytes of memory and an RK05 drive with ~2.5 MBytes capacity.

Routines at that time were not compiled but interpreted.
All .obj,.int,.mc.,cls, ... showed up much later when a compiler was developed.
Actually .int routines describe best what was the unique format at that time.

Writing a routine was essentially writing lines of code into your partition and then saving it on disk.
The related commands were

- PRINT to show a routine on terminal
- ZLOAD to fetch a routine from disk into your partition
- ZSAVE to save a routine on disk
- ZINSERT to insert a single line into a routine in your partition
- ZREMOVE to remove lines from your routine in your partition
- $TEXT() to read a line of your routine into a variable

They already existed and they survived until today also in the latest IRIS versions.

As you had to use your partition to compose your routine there was no other place
to run some program for editing your code. You had to edit it from the terminal prompt
by endless sequences of ZRemove, ZInsert intermixed with Print.

CRT-Terminals at that times had 20 rows by 72 columns (VT05)
- then 12 rows and 80 columns (VT50)
- then 24 rows and 80 columns (VT52), (VT100), (VT220)
None of them had something like a scroll buffer.
Display of line 24 made line 1 go away as you still required an input line.
In addition to remember: there was no Copy/Paste available yet.

So changes of a routine of more than 23 lines were almost impossible without a print-out.
And still then a quite typing intensive exercise.

BUT:

There was an editor that allowed shifting of lines, in-line editing, ...
Just all you might expect from an average routine editor.
To achieve this without destroying the content of your partition its
whole code was just run by XECUTE commands (with no parameter passing at that time)
The trick is that all commands come from variables and never touch the routine loaded locally.

The editor was started by **X**ECUTE **^%** and was homed in namespace %SYS.
And that was my big surprise:
It is still there and operational --- also in my latest version of IRIS

I was almost incredible to see that this tool from the first days is still there and still works as in the past.
I find it remarkable especially with our technology to have such a long lifecycle.

Well, it has its limits:
- It just operates on .INT routines.
- it doesn't run in WebTerm as there is just no free partition to load your routine
- its input displays sometimes in the middle of your terminal if you have more than 24 lines visible.
  This is just a limit set in your terminal configuration.
  Set it to 99 lines and input is always on end of screen.

I'm pretty sure the tool is rather unknown, so here are the available functions:

```
Edit: ?
Enter one of the following
.B   to break a line into two pieces
.C   to change all occurrences of a string
.D   to display s specific range of lines
.F   to file the routine
.FX  to file the routine and exit the editor
.G   to get lines that were put into in a buffer (see .P)
.I   to insert additional lines (or use the TAB key)
.J   to join a line with the next one
.M   to move lines within the routine
.P   to put lines into a buffer (see .G)
.R   to remove one or more lines
.S   to search for all occurrences of a string
tag OR tag+offset  to edit a line
+n   to move down n lines and edit
-n   to move up n lines and edit
""+n to edit the nth line of the program
^globalref  to edit a value stored in a global node
any line of Cache ObjectScript source code, to execute it
     (but if you call another routine before filing this one,
     it will wipe out any edits you have made).
"." use decimal point to move character by character on the current line
" " use space bar to move from space to space on the line
"e" to enter characters, use "." and " " to move to entry point,
    then press "e".  To get out of entry mode, press <RETURN>.
"d" to delete the characters to the end of the word the cursor is on
Backspace: Depending on your terminal settings, backspace will either move
           back character by character, or delete the character at the cursor.
Edit:
```

BTW:
Reading and interpreting the code in global **^%** is an excellent training to understand what ObjectScript does.

#ObjectScript #Tools #Caché #Ensemble #InterSystems IRIS
160 1 0 5 203

Log in or sign up to continue
Add reply

Source URL: https://community.intersystems.com/post/editor-archaeology