

## Column: War Stories

## Figures:

## File for Subscriber Downloads:

# Unforgiven

Whil Hentzen

There is a Visual FoxPro report running in the Rocky Mountains.

It takes 1 ½ minutes to execute, and users are ecstatic. The previous version of the report, written in the pseudo code that was hung on to the side of another tool, took nearly 20 minutes to run. As a Fox developer, this is not big news – you’re used to this type of performance. That’s how many of you got your start – compiling an old dBASE III+ application in FoxBase, and blowing the socks off the users as they saw the performance improvement.

No, the ironic part of this story is that this two page report, processing a table with not quite 30,000 records in it, could well be the most inefficient piece of Fox code ever written. A consultant ran the Coverage Profiler on this report. In 90 seconds, VFP cranked through 1.3 million lines of code to produce the two pages of results. Yes, that’s 1.3 *million* lines of code.

You may be wondering just how one could write code that inefficiently. Here’s an example.

One part of the report requires subtotals and totals on several fields. How would you do this? Something like:

```
select sum(field1), sum(field2) ;
  where <condition> ;
  into array aStuff
```

Well, that’s pretty high-tech, so you couldn’t blame them for being a bit more old-fashioned, along these lines:

```
go top
m.Total1 = 0
m.Total2 = 0
scan
  if <condition>
    m.Total1 = m.Total1 + <field1>
    m.Total2 = m.Total2 + <field2>
  endif
endscan
```

But this wasn’t what they did. Instead, here’s the actual code:

```
* get the sum for field 1 if condition is true
go top
m.Total1 = 0
scan
  If <condition>
    m.Total1 = m.Total1 + <field1>
  endif
endscan

* get the sum for field 2 if condition is true
go top
m.Total2 = 0
scan
  If <condition>
    m.Total2 = m.Total2 + <field2>
  endif
endscan
```

Well, it's not the actual code – they didn't put those superfluous lines with the asterisks in there. But otherwise, you can believe your eyes - a separate SCAN for each field. They didn't even put all of the work inside a single SCAN loop. And the rest of code that runs this report carries on the tradition.

My purpose is not to slam the work of the programmer – we're all guilty of less than stellar coding practices here or there, although this is an extreme example (I hope, at least.) No, my point is the subtle message here – Fox cranked through over a million lines of code in a minute and a half.

That's pretty fast. I thought you had to write in assembler to achieve that type of performance. As a result, you can see that Fox is incredibly forgiving. You can perform with the sloppiest of coding practices, and an incredible lack of knowledge, and *you can get away with it*.

Try this with another tool, though, and you'll be slammed up against the basement wall faster than you can click the Run button. So you'd better get used to learning not just how to get it working once, but how to optimize and tweak to achieve maximum performance. With other tools, if you try a stunt like this report, you won't be forgiven.