

Intentionally making the suggestion look nothing like any scripting language, yet understandable enough to get the point across

 devblogs.microsoft.com/oldnewthing/20150707-00

July 7, 2015



Raymond Chen

In an internal peer-to-peer discussion list for an internal tool I'll call *Program Q*, somebody asked,

| How can I query the number of free frobs in every table in my table repository?

I suggested that they could use the command

```
q query-property "*::frops-free"
```

taking advantage of the fact that in Program Q, you can specify a wildcard for the table name to query across all tables.

| Thanks, this looks promising, but my repository has a huge number of tables, so the `q query-property` command refuses to expand the `*` wildcard that much. I can get around this by issuing 26 queries, one for each letter of the alphabet:

```
q query-property "a*::frops-free"
q query-property "b*::frops-free"
q query-property "c*::frops-free"
...
q query-property "z*::frops-free"
```

| Is there a better way to do this?

I replied with some pseudocode.

```
(
  from table in `q list-tables`
  select table + "::frops-free"
) | q query-property @-
```

(The `@` means that it should take the list of properties from a file, and we give `-` as the file name, meaning standard input. Not that it's important because *I completely made this up.*)

A colleague of mine noted that I provided just enough syntax to explain the algorithm clearly, but in a form that cannot be executed in any scripting language, so the user understands that it is just an algorithm that needs to be massaged into something that will actually run.

It's a neat trick when it works. But when it fails, it fails spectacularly. Fortunately, in this case, it worked.

Bonus chatter: For all I know, that's valid PowerShell.

Raymond Chen

Follow

