

# The problem with adding more examples and suggestions to the documentation is that eventually people will stop reading the documentation

 [devblogs.microsoft.com/oldnewthing/20130410-00](http://devblogs.microsoft.com/oldnewthing/20130410-00)

April 10, 2013



Raymond Chen

I am a member of a peer-to-peer discussion group on an internal tool for programmers which we'll call Program Q. Every so often, somebody will get tripped up by smart quotes or en-dashes or ellipses, and they will get an error like

```
C:\> q select table -s "awesome table"
Usage: q select table [-n] [-s] table
Error: Must specify exactly one table.
```

After it is pointed out that they are a victim of Word's auto-conversion of straight quotes to slanted quotes, there will often be a suggestion, "You should treat en-dashes as plain dashes, smart quotes as straight quotes, and fancy-ellipses as three periods."

The people who support Program Q are members of this mailing list, and they explain that unfortunately for Program Q, those characters have been munged by internal processing to the point that when they reach the command line parser, they have been transformed into characters like ô and ö, so the parser doesn't even know that it's dealing with an en-dash or smart-quote or fancy-ellipsis.

Plus, this is a programming tool. Programmers presumably prefer consistent and strict behavior rather than auto-correcting guess-what-I-really-meant behavior. One of the former members of the Program Q support team recalled,

It might be possible to detect potential unintended goofiness and raise an error, but that creates the possibility of false positives, which in turn creates its own set of support issues that are more difficult to troubleshoot and resolve. Sometimes it's better to just let a failure fail at the point of failure rather than trying to be clever.

There was a team that had a script that started up the Program Q server, and if there was a problem starting the server, it restored the databases from a backup. Automated failure recovery, what could possibly go wrong? Well, what happened is that the script decided to auto-restore from a week-old backup and thereby wiped out a week's worth of work. And it turns out that the failure in question was not caused by database corruption in the first place. Oops.

“Well, if you're not going to do auto-correction, at least you should add this explanation to the documentation.”

The people who support Program Q used to take these suggestions to heart, and when somebody said, “You should mention this in the documentation,” they would more than not go ahead and add it to the documentation.

But that merely created a new phenomenon:

I can't get Program Q to create a table. I tried `q create -template awesome_template awesome_table`, but I keep getting the error “Template 'awesome\_template' does not exist in the default namespace. Check that the template exists in the specified location. See 'q help create -template' for more information.” What am I doing wrong?

Um, did you check that the template exists in the specified location?

“No, I haven't. Should I?”

(Facepalm.)

After some troubleshooting, the people on the discussion group determine that the problem was that the template was created in a non-default namespace, so you had to use a full namespace qualifier to specify the template. (I'm totally making this up, I hope you realize. The actual Program Q doesn't have a template-create command. I'm just using this as a fake example for the purpose of storytelling.)

After this all gets straightened out, somebody will mention, “This is explained in the documentation for template creation. Did you read it?”

“I didn't read the documentation because it was too long.”

If you follow one person's suggestion to add more discussion to the documentation, you end up creating problems for all the people who give up on the documentation because it's too long, regardless of how well-organized it is. In other words, sometimes adding

documentation makes things worse. The challenge is to strike a decent balance.

**Pre-emptive snarky comment:** “TL;DR.”

Raymond Chen

**Follow**

