

The fine art of Yak Shaving



I'm often asked just what it means to be "yak shaving". I can't tell you what the true origin of the phrase is, but believe me, I can tell you that I have shaved plenty of yaks in my career.



Hold still there little buddy, this may take a while

Let's take the example of a process that I'm working on which is to take a CSV file of data which is needed to create and update Microsoft Exchange 2010 contact records. It doesn't seem like a big deal, right?

With the Exchange Management Shell it would appear that the world is my oyster. Unfortunately, I find very quickly that I have some hurdles to get over.

Enter the yak.

Problem #1 - The data that I receive is in a format that isn't compatible with the raw data formatting for the native Exchange contact fields. When I import the fields, I have to concatenate a number of fields such as address (in 3 parts, with 1, 2 or 3 fields populated) and phone number (some numbers are international which creates different formatting) and to top it off the names include a "formal" and "informal" name, where we need the informal name if it is present, but otherwise use the formal name. This is more of a logic issue than a true yak though.

Problem #2 - You can't create a Contact record with all fields, but you can't modify the additional fields unless the initial contact exists. I now have to do a check for the contact to see if it is existing, and if not then I must first create it with the mandatory fields using **New-MailContact** command and then run the **Set-Contact** command to add the address, phone and non-mandatory fields. Again, this could be treated as a programming logic issue but it is a decent setback from what could be a simple 2 case process.

Problem #3 - The process runs great from my Exchange server, or from my workstation. Since I need to run this from a centralized machine that does a multitude of automated processing, I now need to create a working environment on that server to provide the framework to run this process.

This is where the yak shaving really peaks. Now I have to schedule a formal change to install the Microsoft Exchange Management Shell and associated dependencies on the job server. Once that is done, I need to schedule it to run my script with specific credentials. Now I have a new account to manage. The final issue is that the script is not signed, so I need to configure the server's execution policy to run as Unrestricted which requires a sign off as a potential vulnerability.

The moral of the story is that about 40-50% of the time spent, and about 70% of the total duration (due to wait times for change management and environment testing) is spent on being able to run this process which was unrelated to the actual business logic and programming time. That my friends is what yak shaving is all about.

The good news is that it works, and the better news is that I'm finalizing the script and will be posting it to my [TechNet Gallery](#) in the next two weeks so I can hopefully save some of you the shaving time!



That's a little better