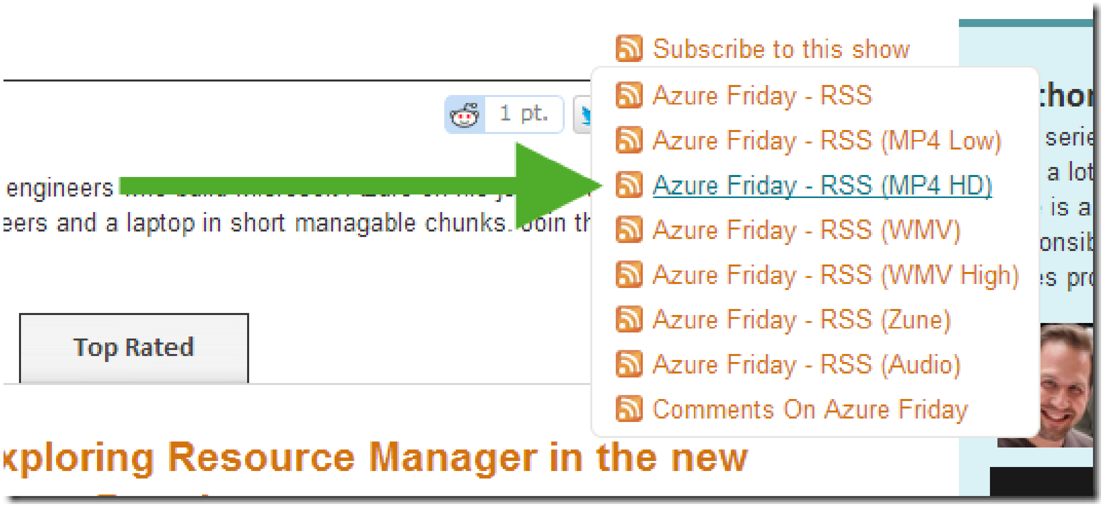
Downloading Azure Friday Videos with PowerShell

23RD OF MAY, 2014 / CHRIS FULSTOW

As it's Friday, I've set myself the challenge of downloading all the Azure Friday videos from the Channel 9 website with a single line of PowerShell. If you're not familiar with this fantastic series, Scott Hanselman talks to the actual engineers who build Microsoft Azure to discuss cloud development in small bite-sized episodes. It's great, and soon you'll be able to watch it offline from the comfort of your own mobile device.
Here goes.

First, I'll need to get a catalogue of all the video titles and their corresponding download links. I could scrape and shred this data, from either the friday.azure.comor Channel 9 websites, using an HTML parser like Html Agility Pack or CsQuery. But that's a lot of effort. Far better to use one of the many conveniently available RSS feeds.

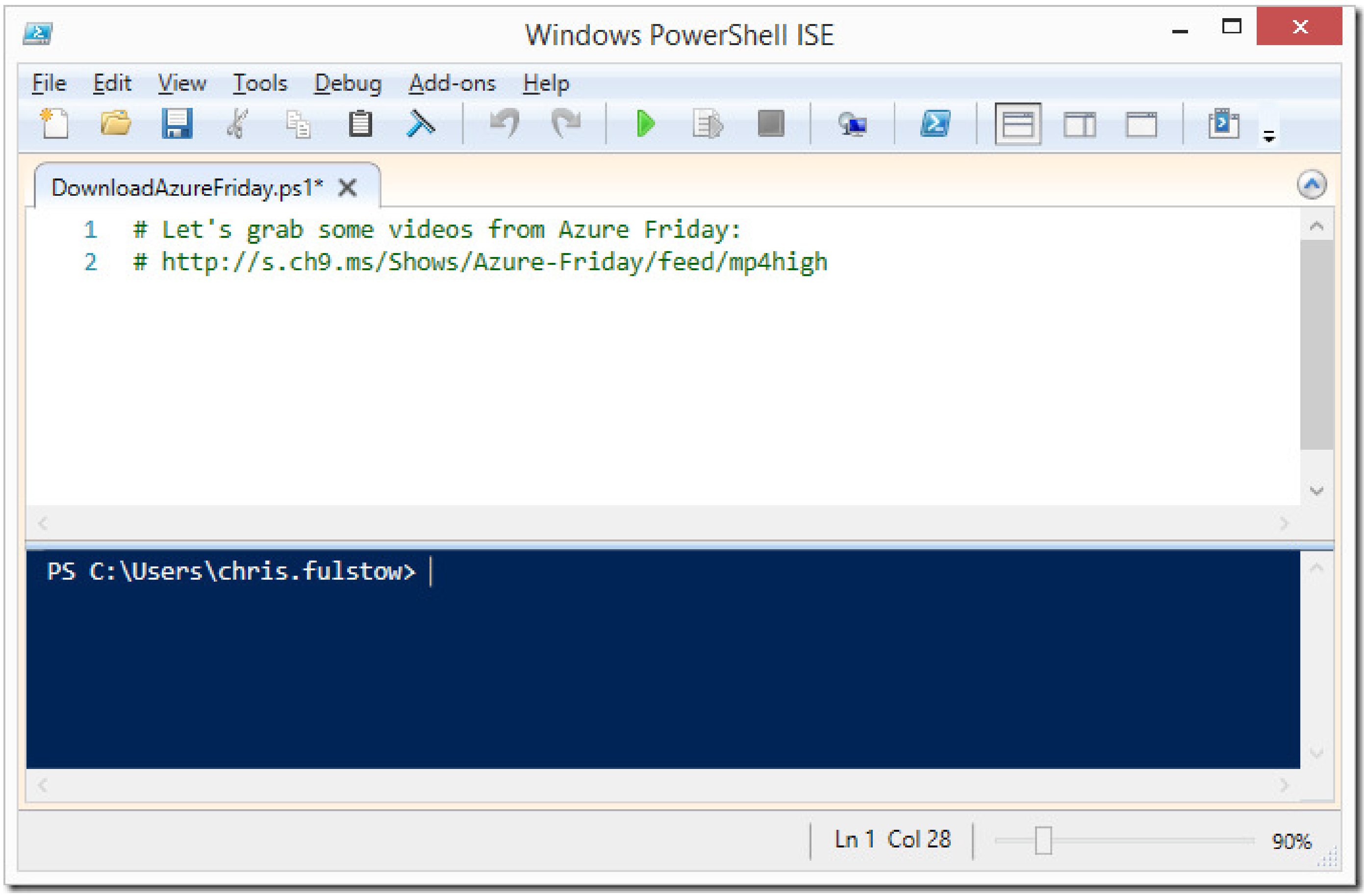

Subscribe to this show
Azure Friday - RSS
Azure Friday - RSS (MP4 Low)
Azure Friday - RSS (MP4 HD)
Azure Friday - RSS (WMV)
Azure Friday - RSS (WMV High)
Azure Friday - RSS (Zune)
Azure Friday - RSS (Audio)
Comments On Azure Friday
Top Rated
1 pt.

I've picked the High Quality MP4 HD feed which has all the information I need, video titles and download links.

Here's what the the RSS XML looks like with extraneous bits removed.

```
<rss>
  <channel>
       <title>Azure Friday (HD) - Channel 9</title>
    <item>
        <title>Exploring Azure Resource Manager</title>
        <enclosure url="http://media.ch9.ms/video.mp4"</enclosure>
    </item>
    <item>
      <!-- More -->
    </item>
  </channel>
</rss>
```

Now I've got all the data I need, it's time to start writing some PowerShell, so I'll fire up the PowerShell ISE and create a new script file.



```
# Let's grab some videos from Azure Friday:
# http://s.ch9.ms/Shows/Azure-Friday/feed/mp4high
```

```
PS C:\Users\chris.fulstow>
```

The first challenge, I've no idea how to download a file over HTTP in PowerShell. There must be a cmdlet somewhere that takes a Uri parameter to download content from the web. Let's take a look.

```
PS> Get-Help * -Parameter Uri | select Name
Name
----
New-WebServiceProxy
Invoke-RestMethod
Invoke-WebRequest
```

That'll do nicely, in fact there's two almost identical cmdlets to choose from, but I'll go with Invoke-RestMethod because, according to the documentation, it automatically deserialises XML response content into objects. That should make it easier later to extract the video title and download link from the XML.
Next, because this script is going to be a pithy one-liner, I want to find whether there's a shorthand alias for Invoke-RestMethod.

```
PS> Get-Alias -Definition Invoke-RestMethod | select Name
Name
----
irm
```

And there is, the not entirely unguessable *irm*, which downloads the RSS and returns it as a nicely formatted array of XmlElement objects. From these, I'll extract the video link and create my filename by removing non-alphanumeric characters from the title.

```
irm "http://s.ch9.ms/Shows/Azure-Friday/feed/mp4high" | `
select `
    @{ n="name"; e={ ($_.title -replace "\W+", '-') + ".mp4" } },
    @{ n="link"; e={ $_.enclosure.url } }
```

Finally, I'll need to download each video and save it to disk. For that, I'll pipe my array of name/link hashes into a Foreach-Object (%) loop that calls Invoke-WebRequest (iwr) and we're done. Here's the finished PowerShell script.

```
irm "http://s.ch9.ms/Shows/Azure-Friday/feed/mp4high" | `
    select `
        @{ n="name"; e={ ($_.title -replace "\W+", '-') + ".mp4" } },
        @{ n="link"; e={ $_.enclosure.url } } | `
    % { `
        iwr -Uri $_.link -OutFile (join-path "D:\Videos\" $_.name) `
    }
```

However, there's one small glitch – the RSS feed only returns the most recent 25 videos. To fetch the next batch of 25 items the RSS feed accepts a *?page=n*querystring parameter, so I could add a loop that increments a page counter until the feed returns no items. Maybe once I've watched these 25 I'll go back and modify the script to grab the rest…