**Search-SPIndex - PSSQT – Search Query Tool – For PowerShell**

PSSQT is a PowerShell wrapper around the search query tool. (https://sp2013searchtool.codeplex.com/)

**Installation**

If you run Windows 10 or have WMF 5 installed (<https://www.microsoft.com/en-us/download/details.aspx?id=50395>), you can very easily install it by running the command

Install-Module -Name PSSQT

directly in PowerShell, and it will be installed from the PowerShell gallery. (<https://www.powershellgallery.com/packages/PSSQT/1.1>)

(There is no need to download the search query tool zip file, if all you want is to install the PowerShell module)

Once it has been installed on your system, you load it by running the command

Import-Module PSSQT

It exports only one cmdlet at the moment: Search-SPIndex

It is primarily a tool for scripting and testing. E.g. automated testing of result sets against a golden set etc.

If you downloaded the search query tool from CodePlex and extracted the zip file, you can load the module directly by running the command

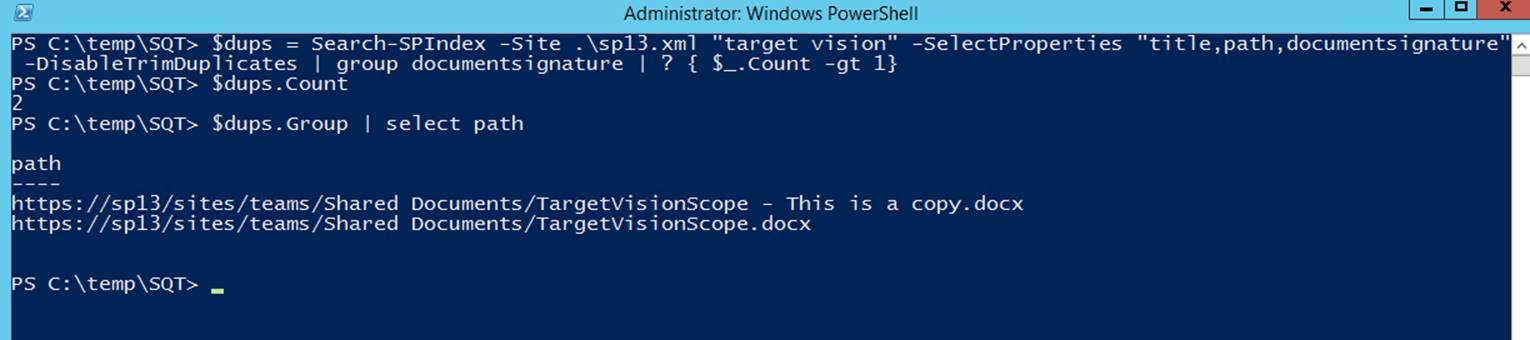
Import-Module <path to search query tool>\psmodules\PSSQT.psd1

E.g:

Import-Module C:\SearchQueryTool2.5\psmodules\PSSQT.psd1

**Some examples for how you can use the tool.**

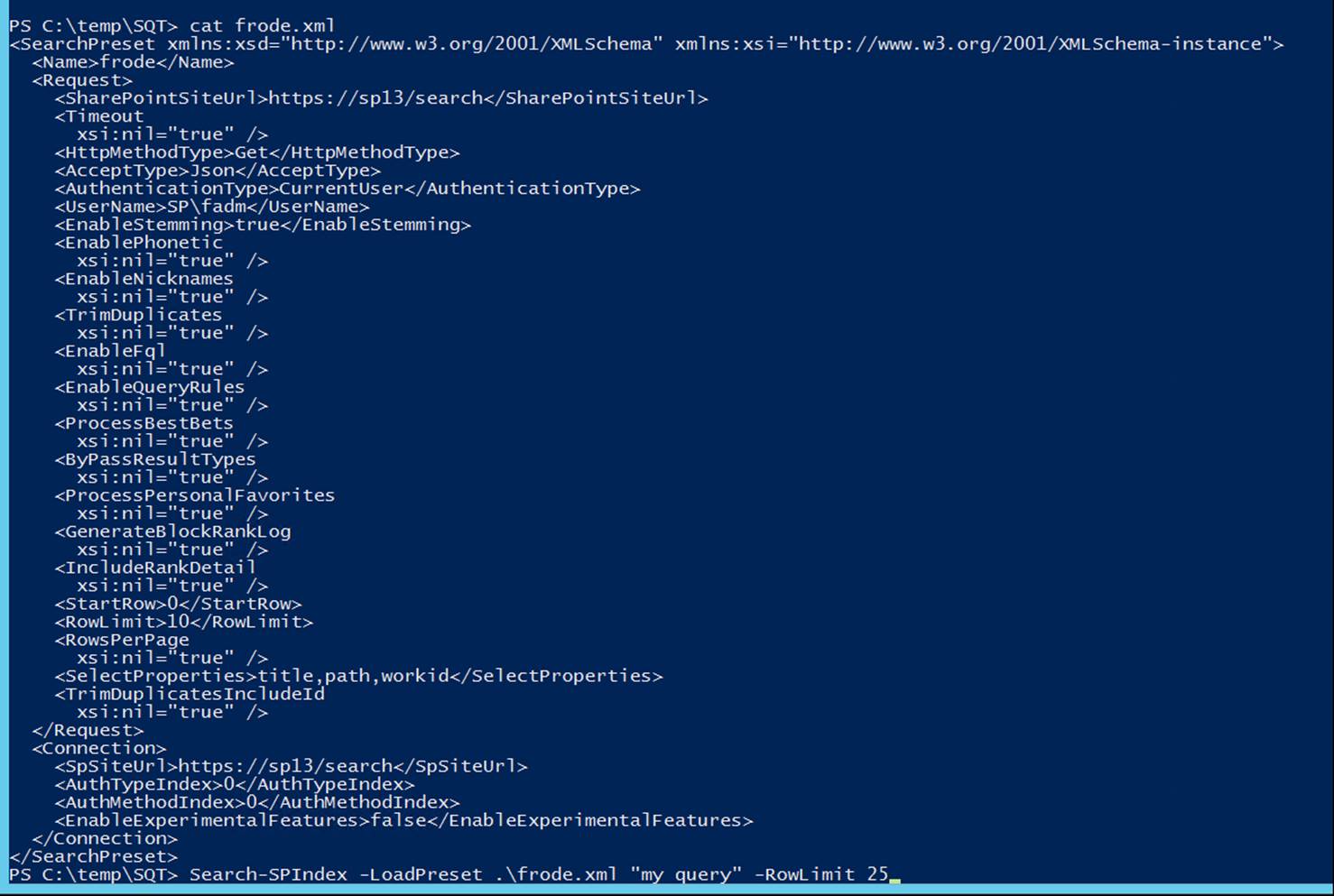
If you want to find duplicates in your index, you can try this:



We search using a query, select the documentsignature property and use PowerShell to group by that and list only entries which has a count greater than 1.

**Presets**

If you use the Search Query Tool UI, you can save presets. PSSQT can use the same preset files. You can also save preset files directly using PSSQT and the -SavePreset switch. When you search using a preset file, you can still override whatever parameter you like on the command line.



**Properties**

You can select properties by using the -Properties command line switch. E.g:

Search-SPIndex -Site <https://sp13/search> -Properties “title,author”

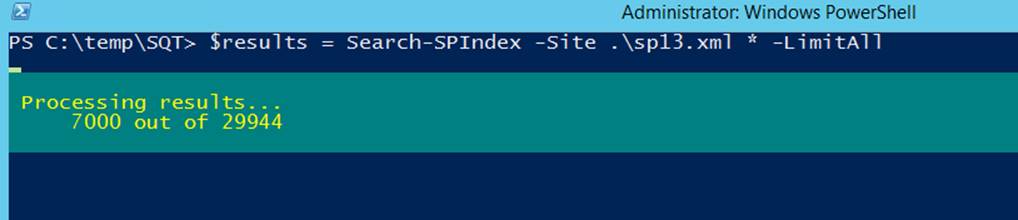
Note: Try this:

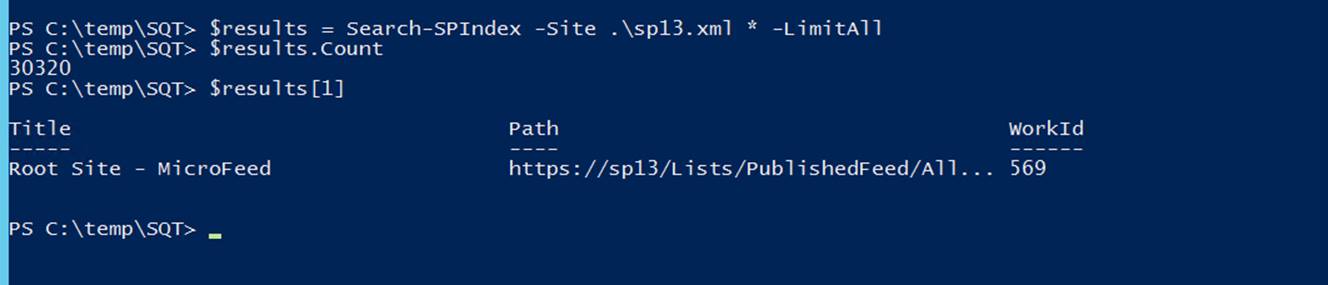
Search-SPIndex -Site <https://sp13/search> -Properties “”

If you use an empty string, it will use a default set of properties.

**RowLimit**

You can set RowLimit to whatever you want (it will batch request in chunks of 500) or you can use -LimitAll (index size and memory permitting. There might be a 100,000 limit as well, but that has never been tested.)

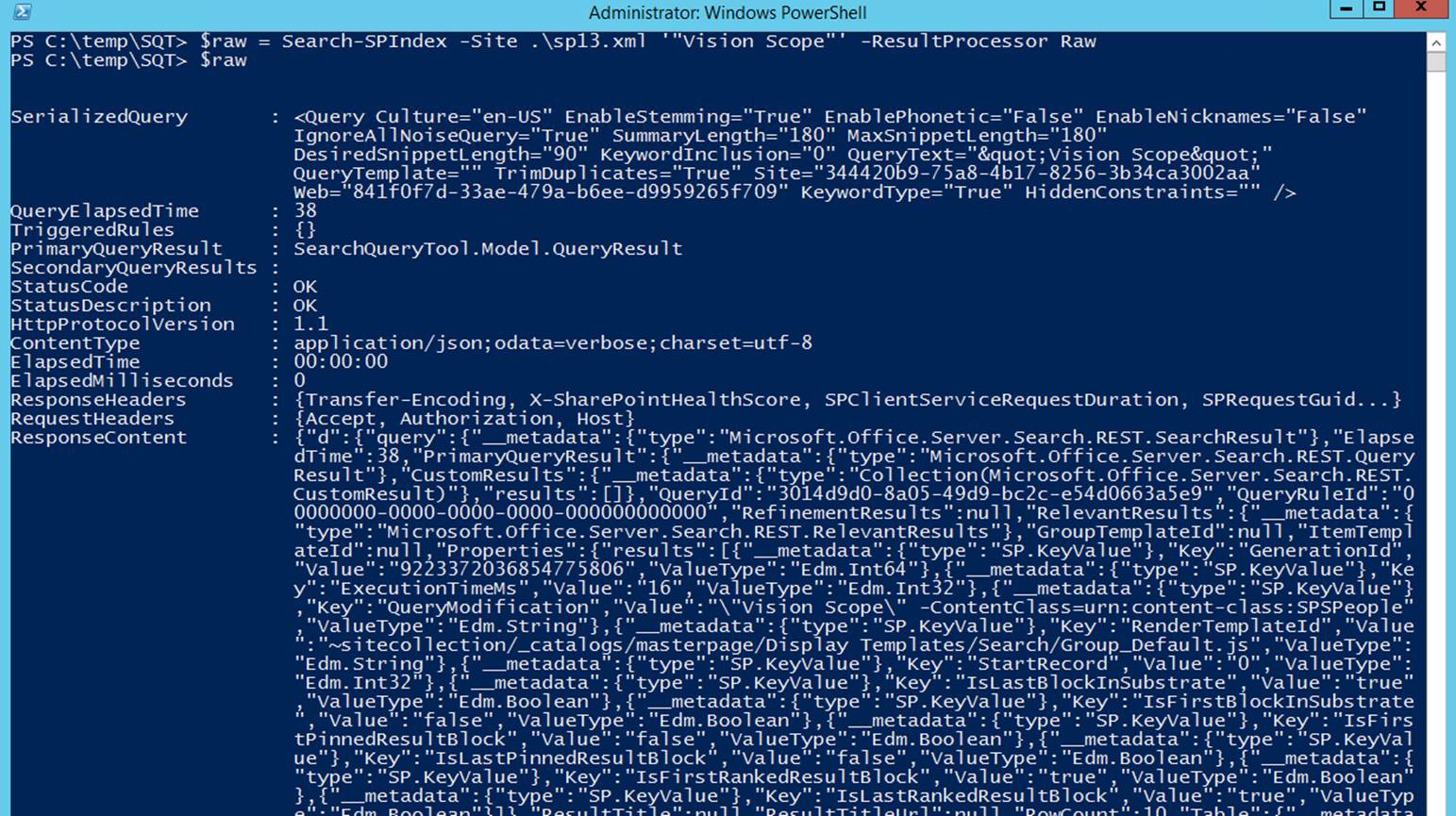




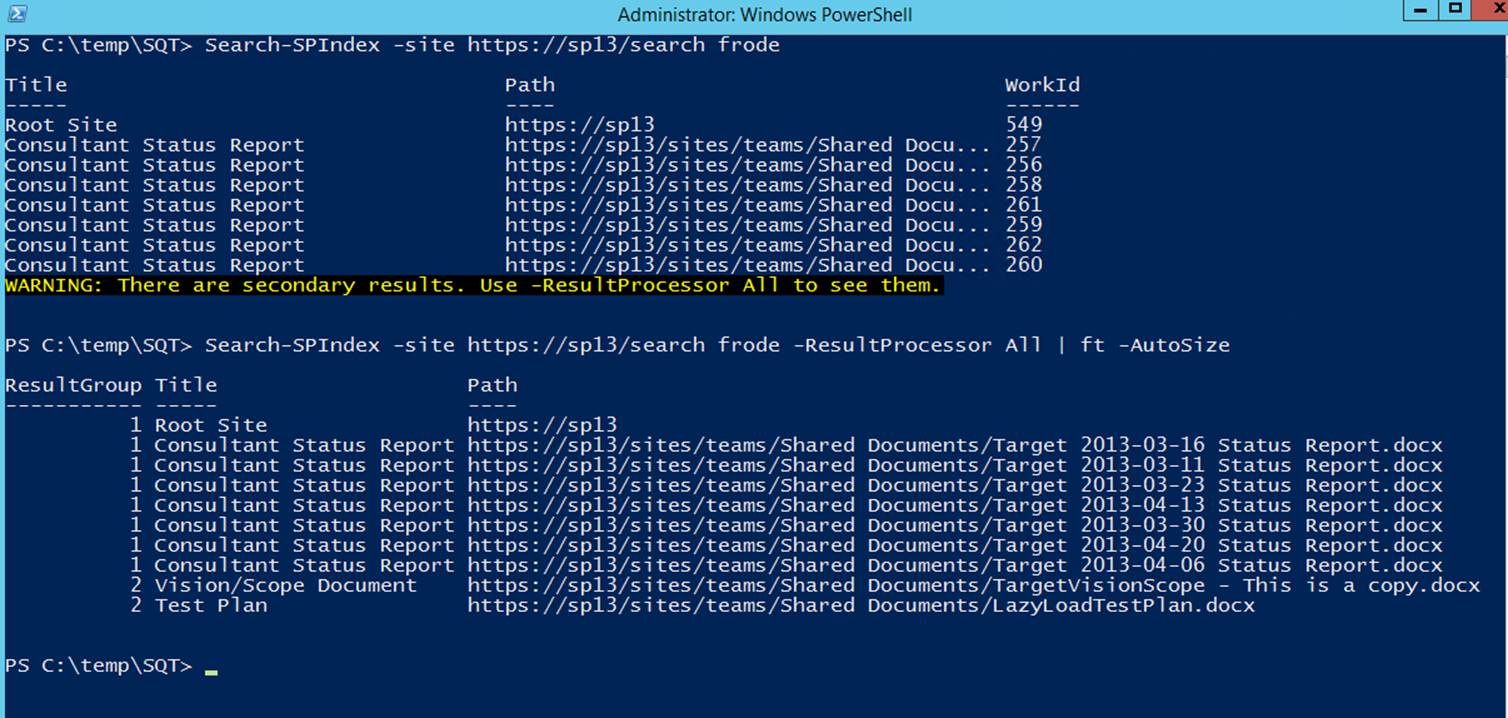
**ResultProcessor**

PSSQT can process the results in different ways by using different ResultProcessors. Use Get-Help Search-SPIndex to see a list of available ResultProcessors.

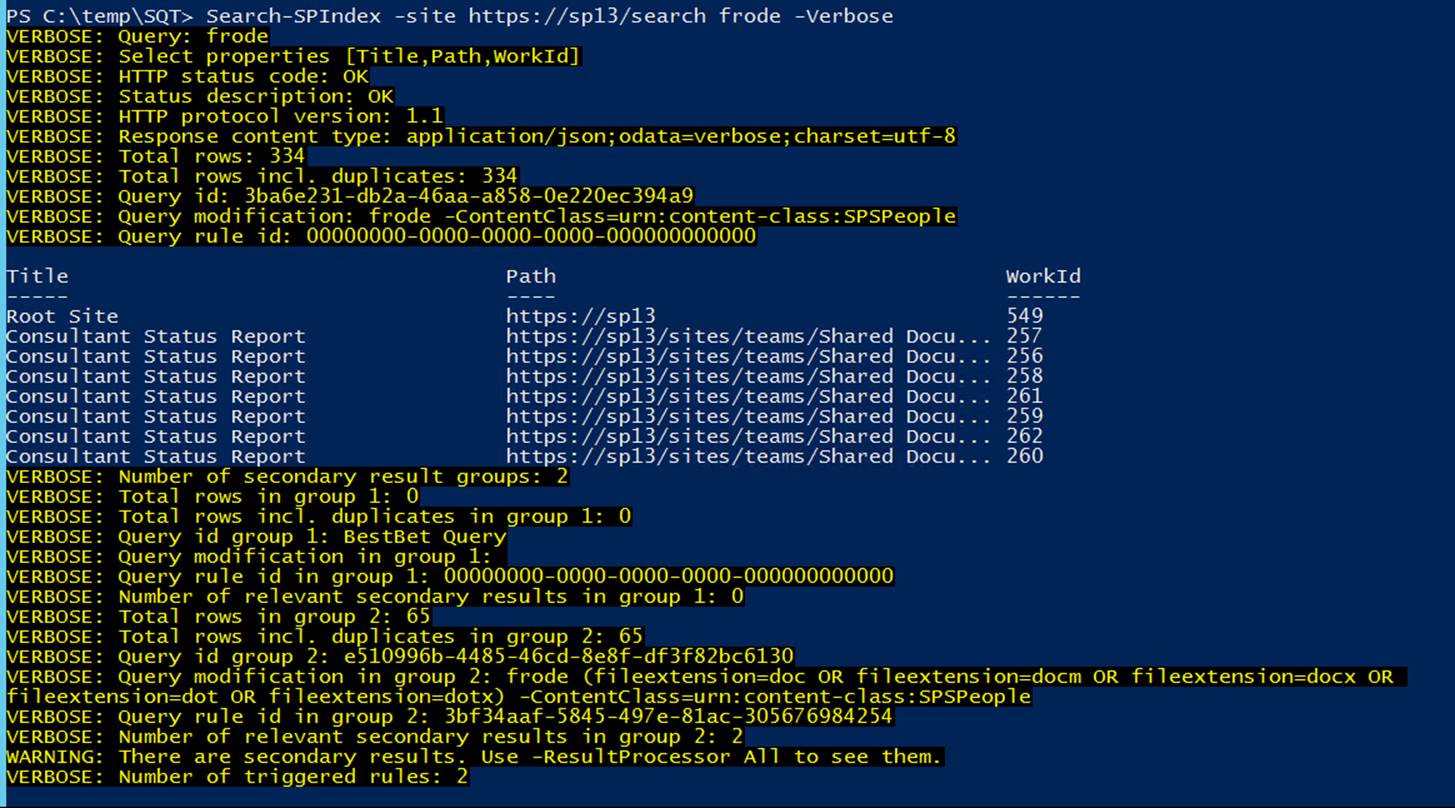
You can get the raw results as well and use the power of PowerShell to slice and dice.



You can get secondary results, refiners, some rank detail etc. Try it out.

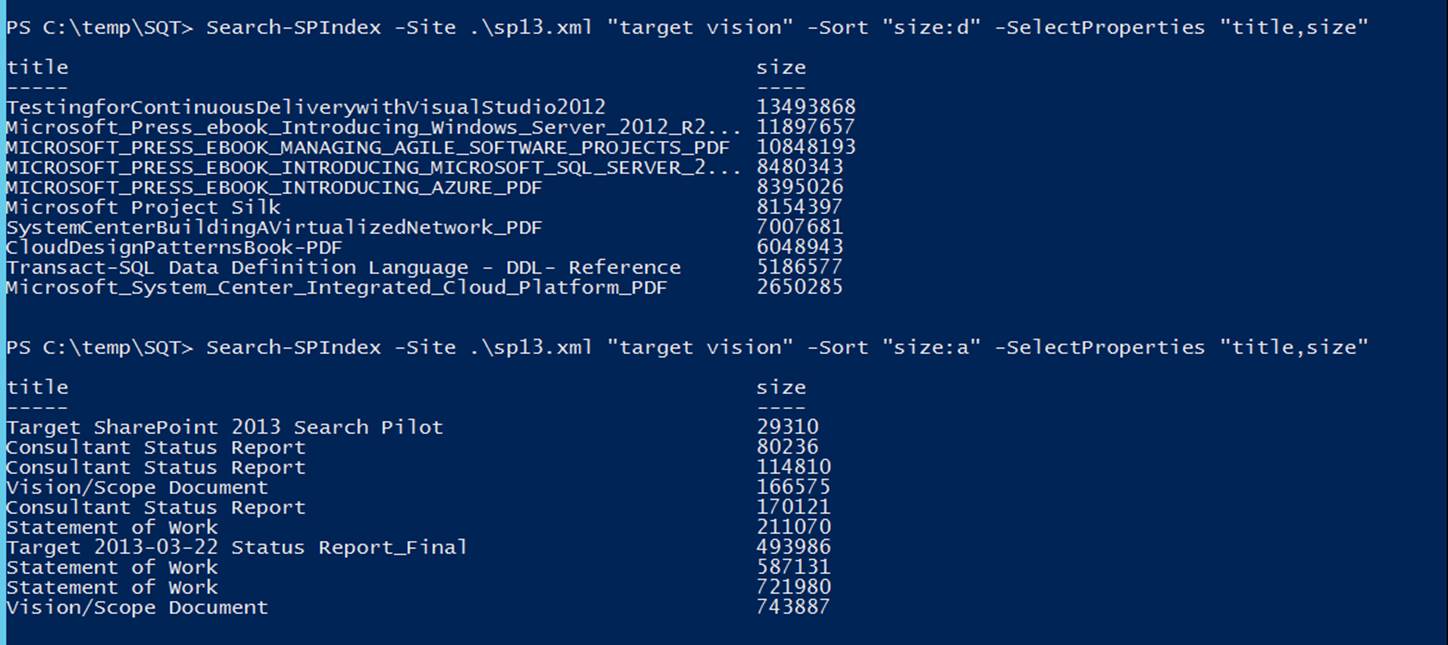


Use -verbose to get a quick overview of more detailed information



**Sorting**

You can specify SharePoint sort property and ascending or descending. You can of course also use PowerShell to sort the results independently. SharePoint of course sorts across the entire index, whereas PowerShell only sorts the returned results.



**Crawling**

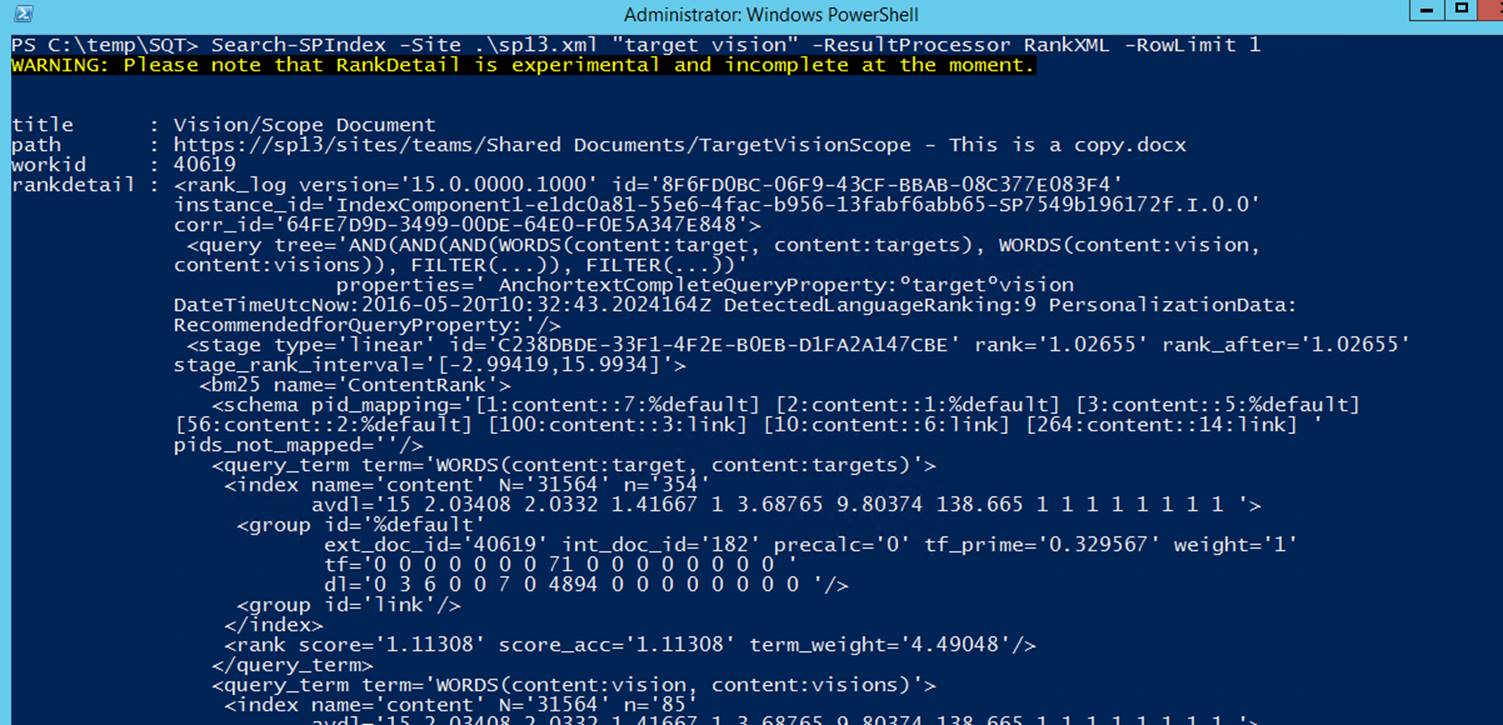
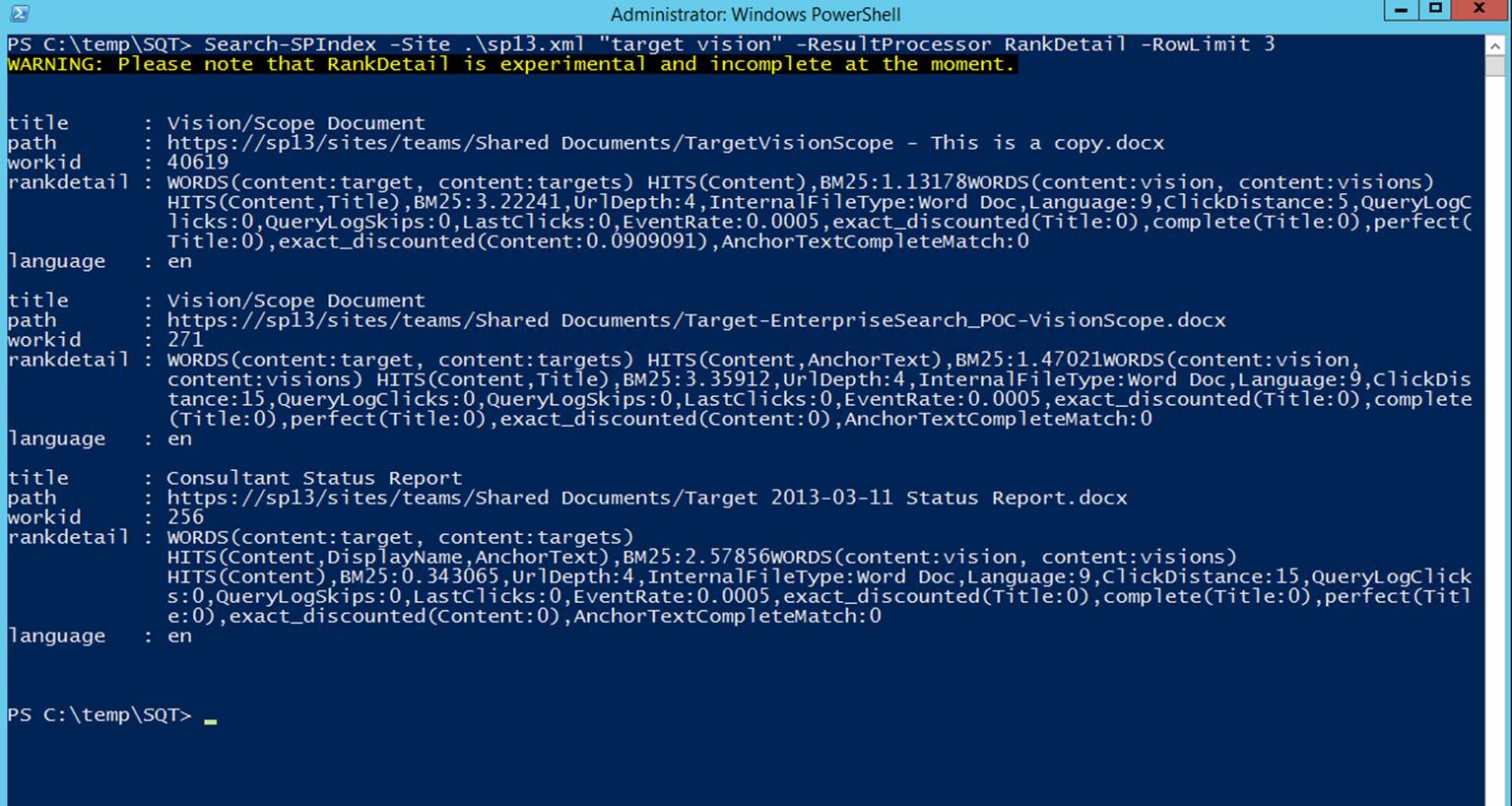
Just an example of how it can be used. Search for a subset of documents and mark them for re-crawl.

$log = New-Object Microsoft.Office.Server.Search.Administration.CrawlLog $ssa

Search-SPIndex -Site https://sp13/search owstaxidGeography:cc07f736-59db-4102-a120-422bb658d5e8 -LimitAll | select path | % { $log.RecrawlDocument($\_) }

**RankInformation**

You can get some very quick and dirty information about RankDetail (or RankXML if you want to use PowerShell to slice and dice) (and it will resubmit query and filter down to 100 results if necessary)



**ClientType**

You can specify client type, but by default PSSQT uses “ContentSearchRegular” to avoid populating the page impression tables in SharePoint with these queries.

