Created by Justin Brazil

https://github.com/justin-brazil/SkryeScript

GPL v3

About

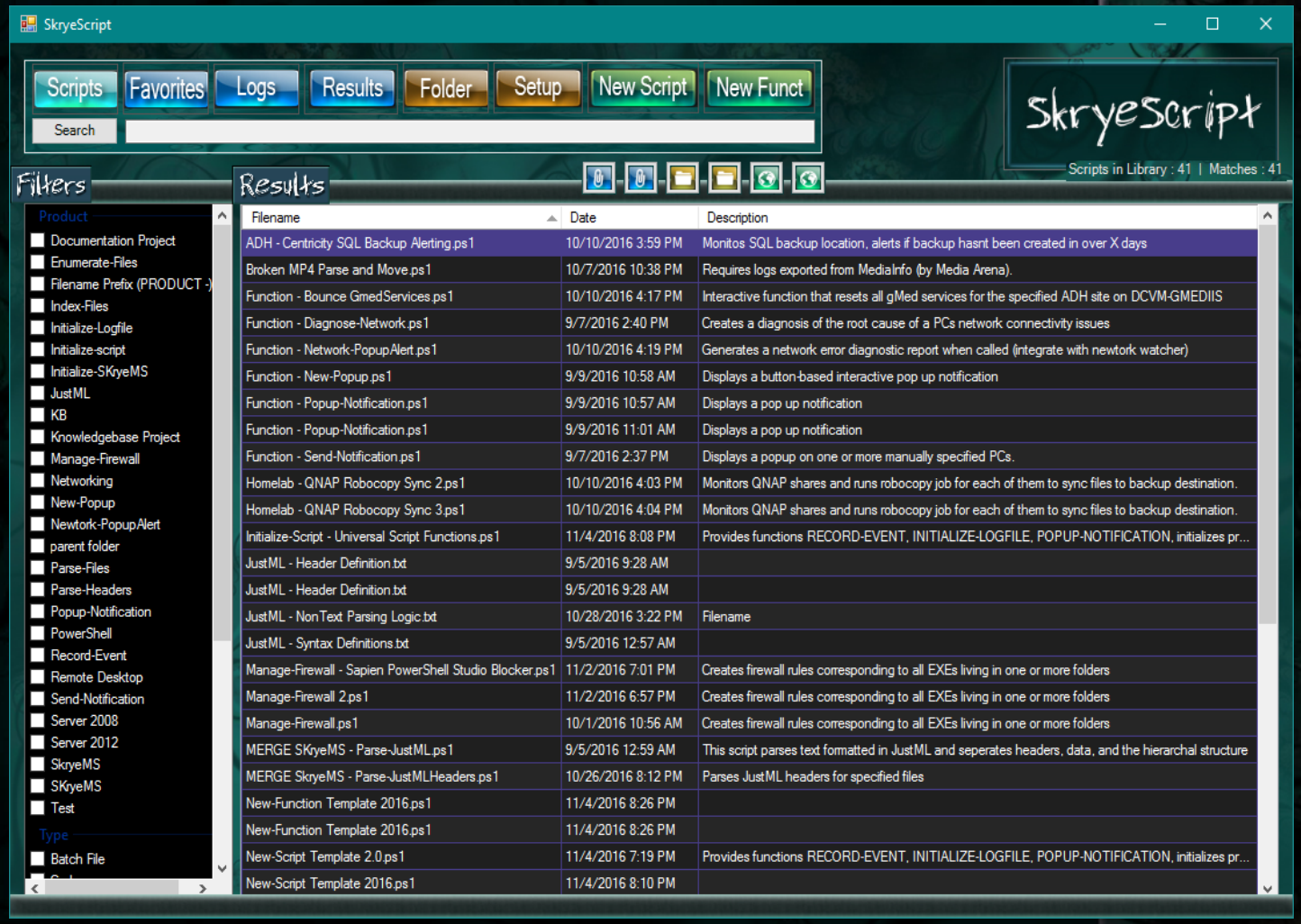
SkryeScript

A free and open-source PowerShell GUI tool designed to manage libraries of scripts and/or text files, with a number of features designed to streamline the scripting process.

Targeted at IT professionals .

Available as an executable or as a .PS1 script, GPL v3 license

Built using PowerShell 5, GUI built with Sapien's PowerShell Studio 2016



What is It?

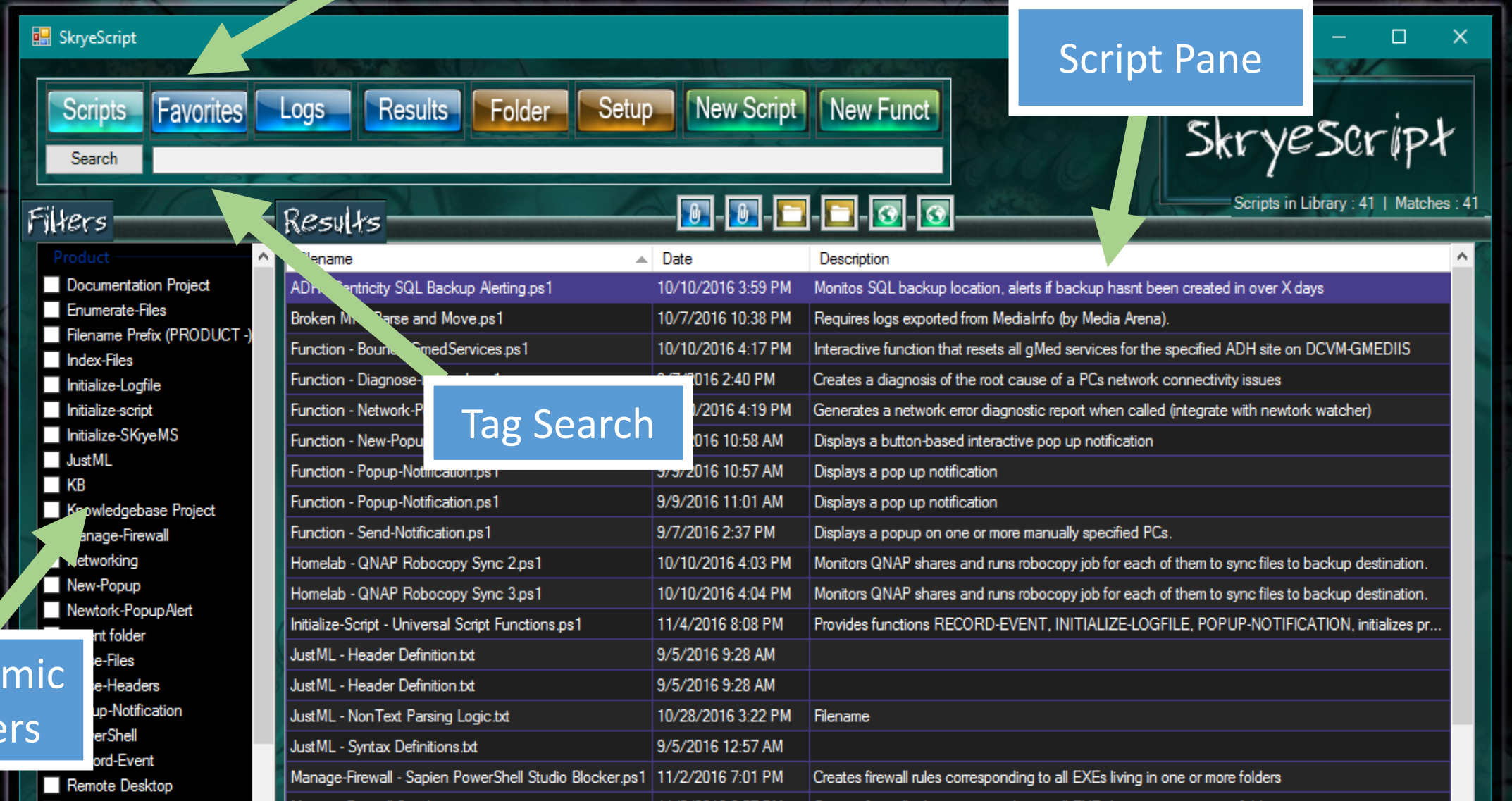
SkryeScript

SkryeScript is a rich GUI tool designed to catalog, tag, and manage text-based files and scripts. It was built by an IT professional for IT professionals with the goal of streamlining the retrieval of script data from various sources, improve scripting workflows, and to standardize logging and output results across all scripts in order to create a consistent environment.

- Creates a library of scripts out of multiple user-defined sources
- Allows tagging, searching, and filtering of indexed items
- 4 selectable modes: All, Favorties, Logs, Results
- Programmable user-defined shortcut buttons, templates, and settings
- Provides standardized frameworks for logging, results, and notifications
- Displays your script's output (logs and results)

Main Screen

SkryeScript



Main Screen

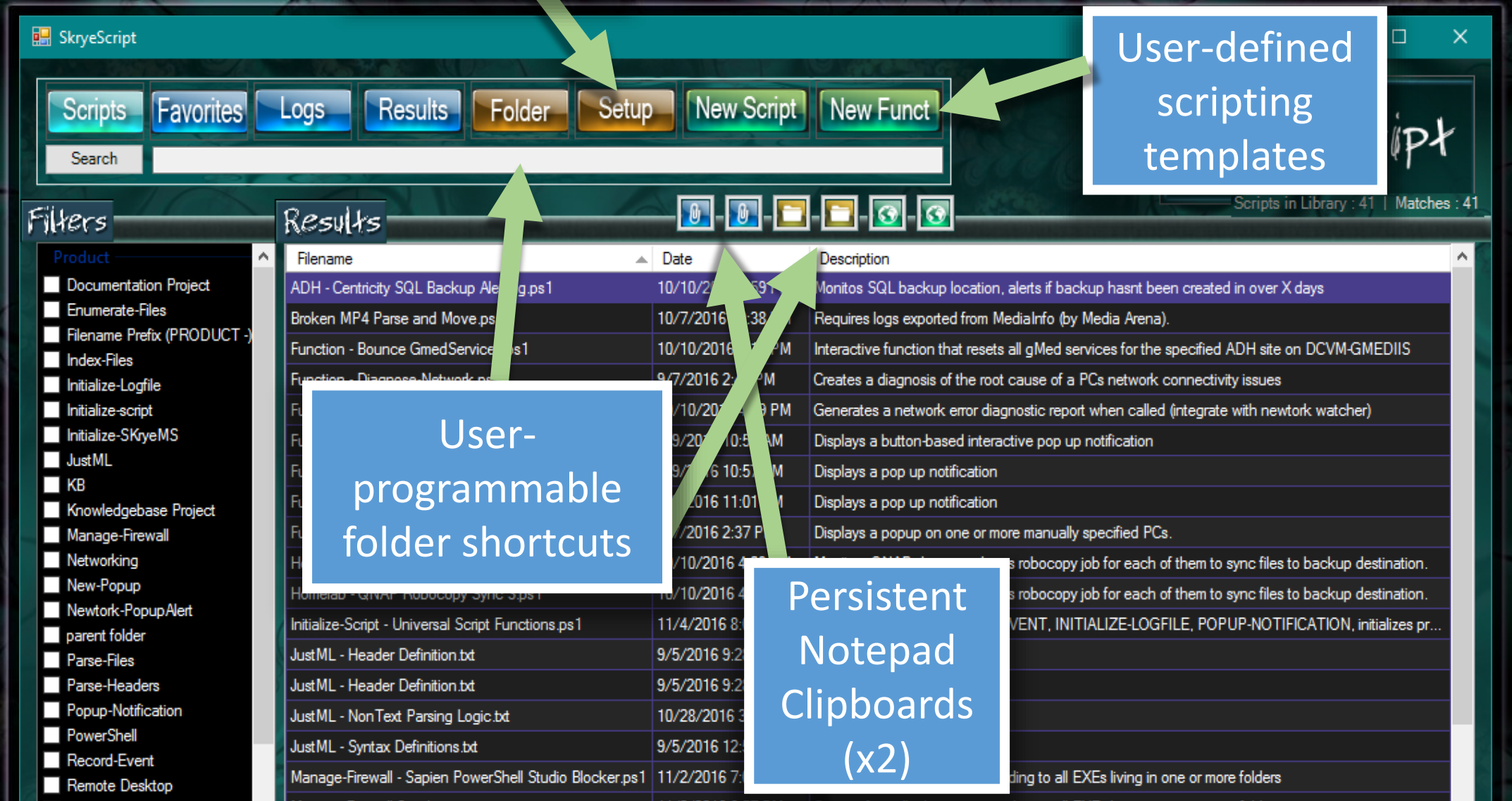
SkryeScript

Setup Menu

User-defined scripting templates

User-programmable folder shortcuts

Persistent Notepad Clipboards (x2)

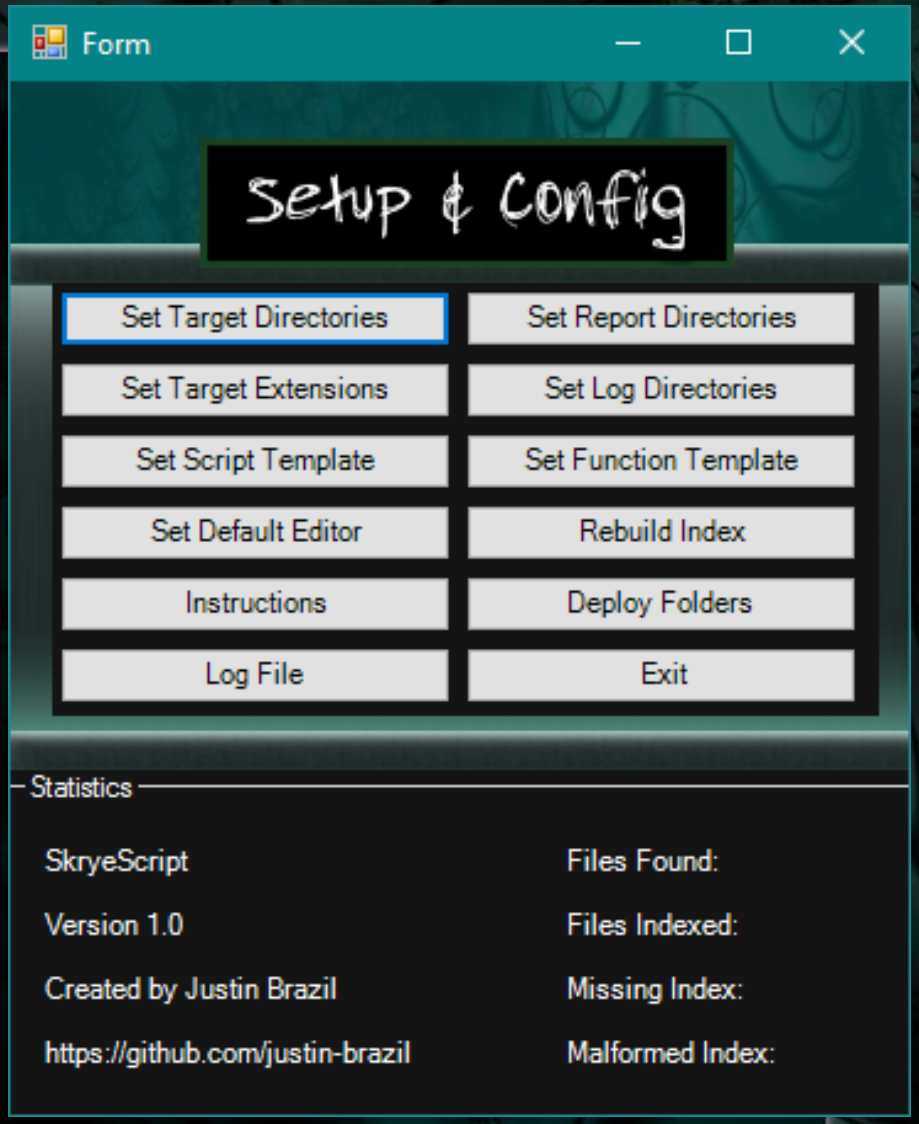


Settings Menu

SkryeScript

Settings are stored in flat-files located in the application's directory and modified via the "Settings" menu.

Customize SkryeMS to suit your preferences and environment.



Beyond the Index...

SkryeScript

SkryeScript is not only an application; it also provides a suggested methodology and a series of standards and templates that can be adopted by anyone who has been struggling with consistency and organization with their scripts.

If you're looking for a tool to get organized with your scripts then check out the suggested standards and the packed "New Script" and "New Function" templates.

If utilized, all logging, results output, script data storage, and disk locations will be handled automatically going forward, and SkryeScript is designed to integrate with these standards to allow you to leverage it as a workflow enhancement tool.


SkryeScript
A Script Organizer for the IT Professional
Indexing Your Data

Script Index

SkryeScript

Builds index of scripts on launch

Scripts indexed using one of three methods:

1) JustML headers (customized markup syntax)
2) #TAGS line added to script
3) Via filename

Method 1 - JustML

SkryeScript

JustML is a custom markup language used to tag text-based data for grouping and parsing and is the most robust option of the three indexing methods. Each script is parsed for special JustML headers that you place at the top of each of your scripts (templates included):

```
#####################SCRIPT_CONTROLS##
#!#Name:    Invoke-Awesome v3.1
#!#Description:    Orders pizza on a schedule via web APIs
#!#Tags:   Pizza,awesome,delivery,API,invoke,hungry
#!#Type:   Function
#!#Product:  PowerShell
#!#Modes:  Scripting,Personal
#!#Notes:   Specify CC# as string
#!#Link:    www.pizzahut.com
#!#Group:   Fun,Office
#!#Special:  Favorites
####################/SCRIPT_CONTROLS##
```

Method 2 - #TAGS

Skryescript

Simply add a #TAGS line near the top of your scripts in order to create searchable/filterable tags.

This method works well but is much less robust than JustML headers.

```
#TAGS Function,Hardware,Audit,Specs,OS,Disk,NIC,Full,Report,Custom,Scan
```

Method 3 – File Name Parsing

SkryeScript

File name parsing can be used in lieu of header-based data. This method works for searching and sorting but lacks the flexibility and power of the header-based methods.

Files are parsed via the following conventions:

C:\KB\Dell\Compellent - Compellent Training (Documentation-SAN,Certification,Training).pdf

Tagging Your Data

SkryeScript

Dealing with script sprawl and multiple script locations requires that you adopt some form of organization and methodology. By adopting a framework such as JustML (and using the built-in templates when writing new scripts and functions) your data will be much more accessible, although it does take a bit of effort upfront.

Try adding JustML headers to a small number of files and see how powerful it is. You can use these headers on all scripts you create going forward (via the templates), and parse your existing scripts using the filename method, or sit down and update your existing scripts all at once using the JustML-BulkImport routing to automatically append JustML headers to the scripts in your library (downloaded separately).

A properly cataloged index is a marvelous thing and will serve you for years to come.


SkryeScript
A Script Organizer for the IT Professional
Consistency Through Standards

Applying Standards

SkryeScript

SkryeMS implements a few basic standards for handling common output such as logs, temp files, persistent script data, and results.

These standards are outlined in this section and included inside of the default script/function templates.

If you do not already have standards it is recommended that you begin utilizing these to handle your logging, results output, and notifications for all new scripts in order to ensure consistency when scripting.

Standardized Logging and Alerting

SkryeScript

Once you dive into utilizing PowerShell in a live environment it becomes important to standardize the handling of logging and alerting.

If you have not already adopted a method for handling this, I recommend adopting the functions included in this installer (and accessed via the New Script and New Function template buttons. These functions are easy to use and will handle all logging, alerting, and console output messages in a consistent manner. Furthermore, SkryeScript integrates with these functions via the "Logs" and "Results" pane, enabling you to pull up logs and results from a nicely organized GUI.

DEPLOY-FOLDERS : Deploys/verifies consistent folder structure for all scripts

INITIALIZE-LOGFILE: Automatically creates and names log files using the above folders

RECORD-EVENT: Handles all user notification, logging, and alerting functions

DEPLOY-FOLDERS

SkryeScript

DEPLOY-FOLDERS is a stand-alone function that you call at the start of a script without any parameters.

It will check and create a set of standardized folders on the root of C: to be used by all PowerShell scripts for storing and organizing logs, results, temp files, and persistent application data for each script.

The other functions automatically fit into these folders and generate automatic variables for you to use when defining things like log files and results output files.

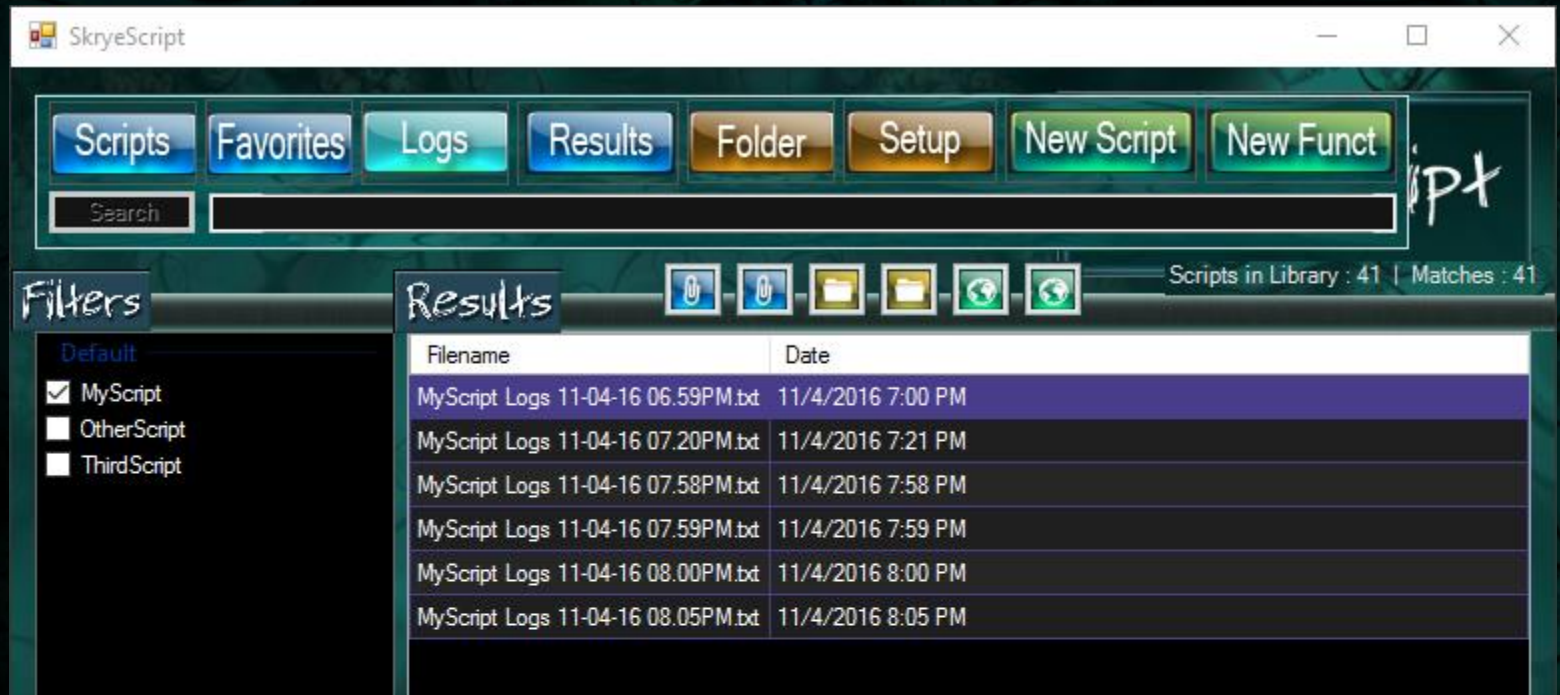
Deploys Folder Structure: C:\PowerShell

INITIALIZE-LOGFILE

SkryeScript

INITIALIZE-LOGFILE : Automatically names and creates log files for your script and stores them in the logs directory created by DEPLOY-FOLDERS.

SkryeScript automatically parses and seperates these results by script when in “Logs” mode.



RECORD-EVENT

SkryeScript

RECORD-EVENT handles all events, notifications, errors, and log-writing using a simple syntax.

RECORD-EVENT supercedes Write-Error, Write-Warning, Write-Host, logging operations, and other methods you may use to communicate data to the user.

SYNTAX: RECORD-EVENT “Your Message Here” –YourSwitch

```
RECORD-EVENT "File Scan" -SectionStart
RECORD-EVENT "Unable to verify selected File" -Warning
RECORD-EVENT "Successfully found $FILE.FullName" -Status
RECORD-EVENT "File Scan" -SectionEnd
```

--SWITCHES--

-SectionStart – Writes to the log file and indicates that a section of your script has started
-SectionEnd – Indicates that a section of your script has completed
-Log – Writes message to the logfile only
-Status – Writes status message to host and log file
-Display Displays specified message to PowerShell console, then writes to log file
-Popup Displays a popup notification form to the user .
- TerminatingError Displays message as error in console, writes to log, creates popup notification, terminates script
- Error Writes error to console and logfile
-Warning Writes warning to console and logfile

Core Features

SkryeScript

Builds a searchable indexed library of script/text resources spanning multiple locations

Powerful search, filtering, and tagging mechanisms allow you to organize and manage large script libraries

Log and Results parsing via standardized interface

Customizable interface designed to streamline scripting workflows, including user-defined templates, text editors, programmable bookmarks, and other program settings

Going Forward...

SkryeScript

SkryeScript was designed to be customizable so that others could adapt it to their workflows. It was written by Justin Brazil – a Systems Architect by trade who enjoys playing with PowerShell - as an experiment in PoSH-driven GUI tooling.

Community feedback and involvement is most welcome – if you have ideas for improvements please dive in.

Regards,
Justin