

Make PowerShell A Real Solution In 5 Steps

Automate. Delegate. Relax.

Heiko Brenn  @heikobrenn

www.scriptrunner.com



Agenda



What you can do with
PowerShell

-
Today: PowerApps &
Power Automate

Automate & Delegate
recurring tasks
with PowerShell &
ScriptRunner



Win An Amazon Gift Card
Raffle at the end of this session

lp.scriptrunner.com/en/powershell-saturday-raffle



Short profile Heiko Brenn



A screenshot of Heiko Brenn's LinkedIn profile. The profile picture shows a man with glasses and a yellow face mask. The background of the profile header features a road leading towards a bright horizon with the text "POSSH HIGHWAY TO SHELL". Below the header, the profile information includes:

Heiko Brenn
@HeikoBrenn
#PowerShell enthusiast @Script_Runner | Music lover | Drummer for @ScriptyBoyz | Co-Founded @PSUGRheinNeckar
⌚ Germany ⌚ scriptrunner.com 📅 Joined September 2018
786 Following 729 Followers

- Product Expert & Head of International Business
- In the IT industry for 25+ years
- Many years of experience as
 - Administrator
 - Consultant
 - Product Manager
- Responsible for EMEA and North America
 - Technical Sales
 - Consulting
 - Partner Management
- Contact information
 - Phone: +49-162-4879156 and +1-6469806339
 - Email: heiko.brenn@scriptrunner.com
 - LinkedIn: www.linkedin.com/in/HeikoBrenn
 - Twitter: twitter.com/heikobrenn
 - GitHub: github.com/HeikoBrenn



Windows PowerShell

www.scriptrunner.com

Condition
if ((Get-Date).Year -le 2014) { "Old" } else { "New" }

Loops
for(\$i = 1; \$i -le 10; \$i++) { \$i }
while(\$i -le 10) { \$i; \$i++ }
do { \$i; \$i++ } while (\$i -le 10)
foreach (\$p in (Get-Process iexpl)



linkedin.com/in/HeikoBrenn

github.com/HeikoBrenn

+49-162-4879156

heiko.brenn@scriptrunner.com

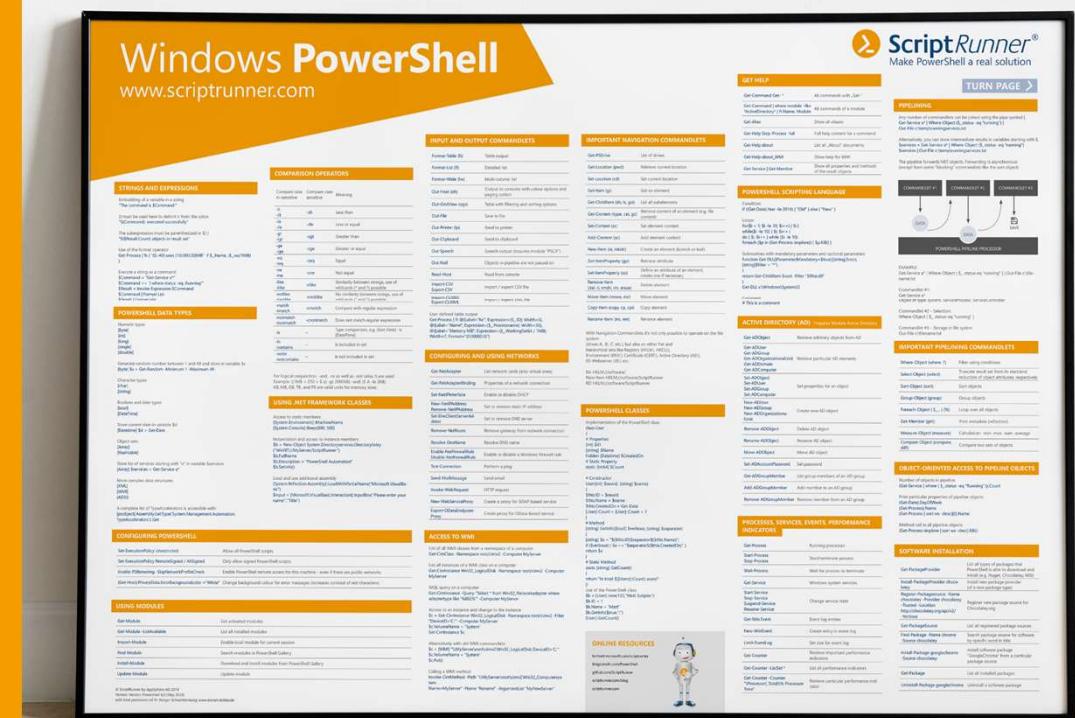


ScriptRunner®

The #1 for PowerShell Management

Get your free PowerShell poster

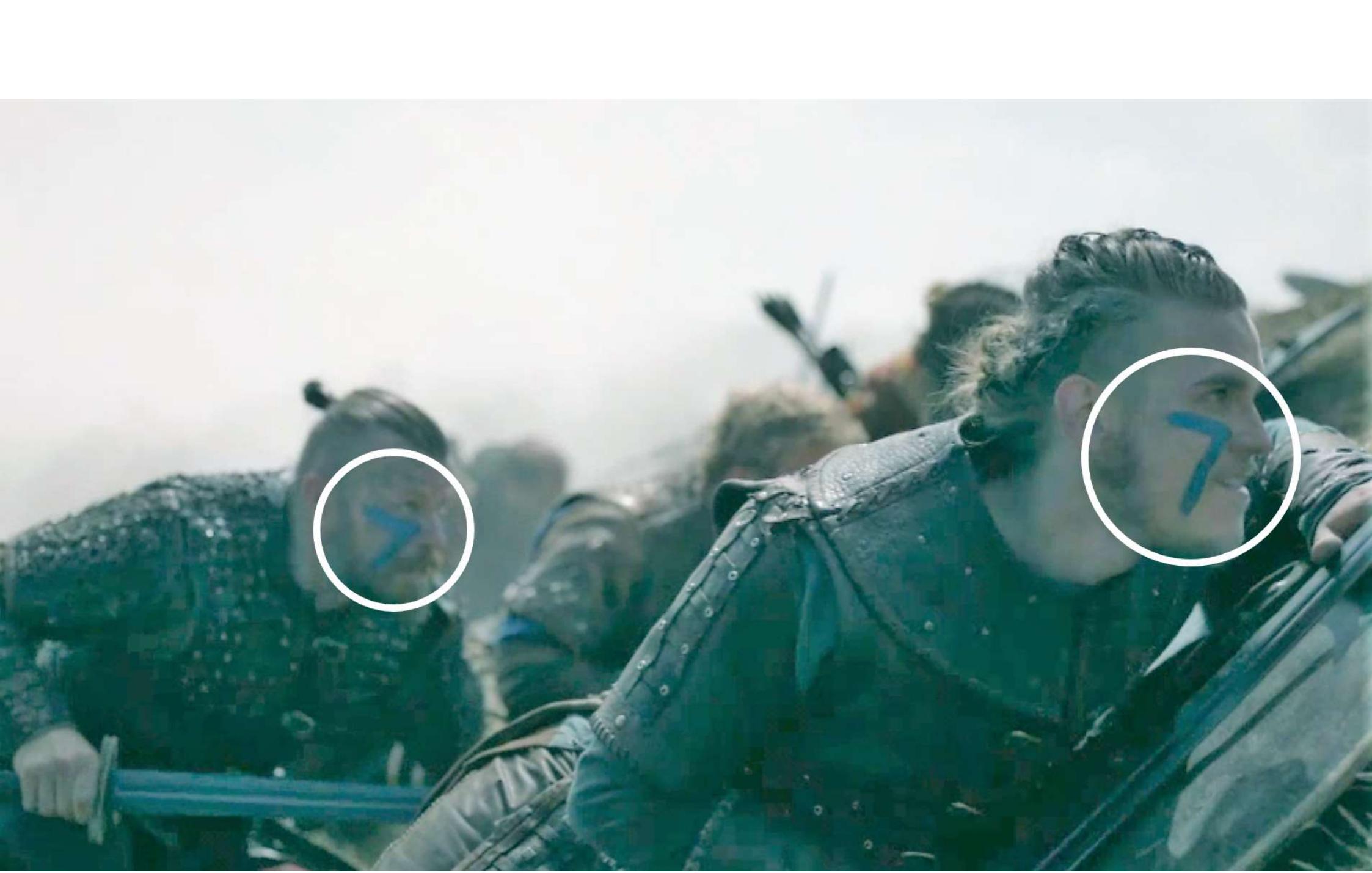
lp.scriptrunner.com/en/powershell-poster



What you can do with PowerShell

Today: PowerApps & Power Automate





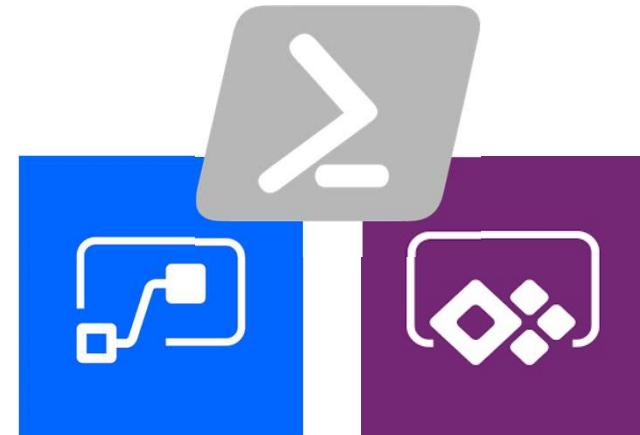
Managing and configuring Flows,
PowerApps, Environments, etc.

Module for administrative features
- Administrator (120 Cmdlets)

Module for configuration features
- Maker (36 Cmdlets)

Module for static analysis checks
- Checker (4 Cmdlets)

More Modules...



Prerequisites



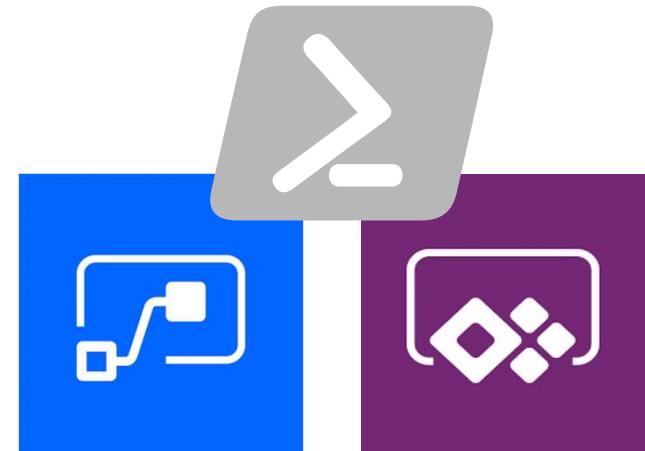
Maker – Valid PowerApps license

Maker – Access to their own resources only

Administrator – Office 365 or Azure AD Global Administrator

Administrator – Dynamics 365 Service Administrator

Administrator – Signed in to PowerApps Admin Center once



Example Use Cases



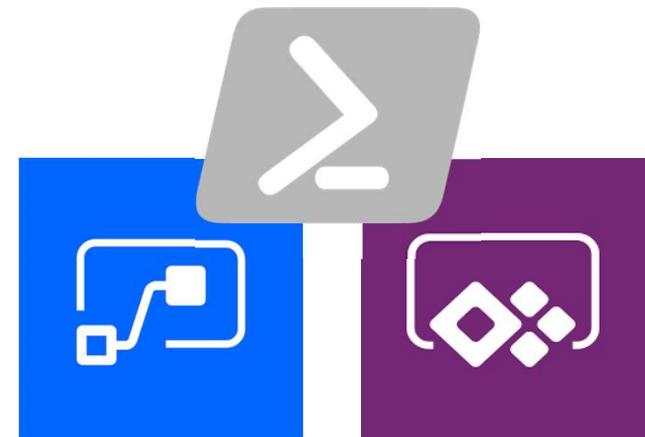
Create/Delete Environments

Add/Remove Users to/from
PowerApps/Flows

Enable/Disable Flows

Get Current Environment/
PowerApps/Flow Settings

Create reports



Demo

PowerShell examples for
PowerApps & Power Automate



Automate & Delegate recurring tasks with PowerShell & ScriptRunner



Does this remind you of the way PowerShell
is used in your organization today?



„Unorganized“ PowerShell usage

PowerShell know-how bottleneck

No org-wide secure
credential management

PowerShell is only used
in expert circles

The potential of PowerShell
remains untapped



What it takes to successfully automate with PowerShell in the enterprise



Standardization

Centralization

Easy accessibility for non-IT experts

Monitoring

Integration



Accessibility for non-IT experts



BEFORE

A screenshot of the Windows PowerShell ISE interface. The title bar says "Administrator: Windows PowerShell ISE". There are two tabs open: "Untitled3.ps1" and "Exchange_Webinar.ps1". The "Untitled3.ps1" tab contains a large block of PowerShell script code. The code is used to set up auto-replies for multiple mailboxes. It uses variables like \$cmdArgs, \$msg, and \$box, and performs operations like Set-MailboxAutoReplyConfiguration and Get-Mailbox. The code is color-coded for syntax highlighting. The status bar at the bottom shows "Completed" and "Ln 1 Col 36".

```
118 $cmdArgs.Add('ExternalAudience', $replyType)
119 $cmdArgs.Add('InternalMessage', $InternalText )
120 $cmdArgs.Add('ExternalMessage', $ExternalText )
121 if($PSCmdlet.ParameterSetName -eq "Schedule Auto Reply"){
122     $cmdArgs.Add('AutoReplyState', 'Scheduled')
123     $cmdArgs.Add('EndDate', $EndDate)
124     $cmdArgs.Add('StartTime', $StartDate)
125     $msg += "scheduled"
126 }
127 else{
128     $cmdArgs.Add('AutoReplyState', 'Enabled')
129     $msg += "enabled"
130 }
131
132 $Script:resHtml = @()
133 foreach($item in $MailboxIds){
134     try{
135         $box = Get-Mailbox -Identity $item -ErrorAction Stop
136         if($null -ne $box){
137             try{
138                 $null = Set-MailboxAutoReplyConfiguration @cmdArgs -Identity $box
139                 if($GenerateReport -eq $true){
140                     $Script:resHtml += Get-MailboxAutoReplyConfiguration -Identity $box
141                 }
142                 $resultMessage += [System.String]::Format($msg,$box.UserPrincipalName)
143             }
144             catch{
145                 Write-Output "Error occurred at set Mailbox $($item)"
146             }
147         }
148     }
149     catch{
150         Write-Output "Error occurred at get Mailbox $($item)"
151     }
152 }
```

AFTER

A screenshot of the ScriptRunner SelfService interface. The top navigation bar shows "1.company.net/OutOfOffice/" and "SelfService". The main page has a title "OutOfOffice" and a subtitle "Create OOF Notifications With ScriptRunner Integration". Below this, there are several input fields:

- "Please select the desired Parameter Set:" dropdown menu showing "Schedule Auto Reply".
- "Please select mailbox(es) Heiko was here:" search bar with results "Angus Young", "Shania Twain", and "Phil Collins".
- "Please select the notification template:" dropdown menu showing "Working From Home|Home Office".
- "Please select the start of absence" date input field showing "Feb 4, 2021, 5:21 PM".
- "Please select the end of absence" date input field showing "Feb 11, 2021, 5:21 PM".

A green play button icon is located in the bottom right corner of the form area.

Centralizing PowerShell



STANDARDIZATION



COLLABORATION



SUPERVISION



SECURITY



DELEGATION



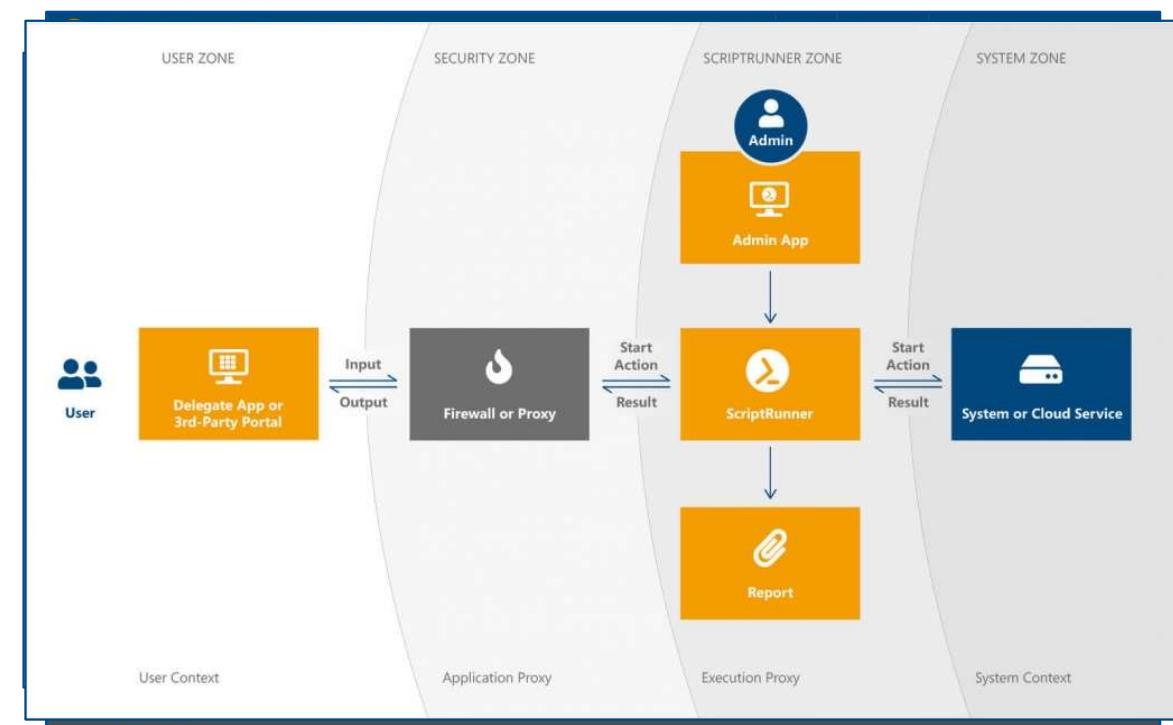
AUTOMATION



Make PowerShell A Real Solution In 5 Steps



-  Centralize your PowerShell scripts and modules
-  Manage credentials & permissions securely
-  Automatically transform PowerShell scripts into Web GUIs
-  Delegate recurring tasks to help desk and end users
-  Integrate with ITSM, Monitoring & Workflow systems and more



Hundreds of free
and ready-to-use
PowerShell scripts.

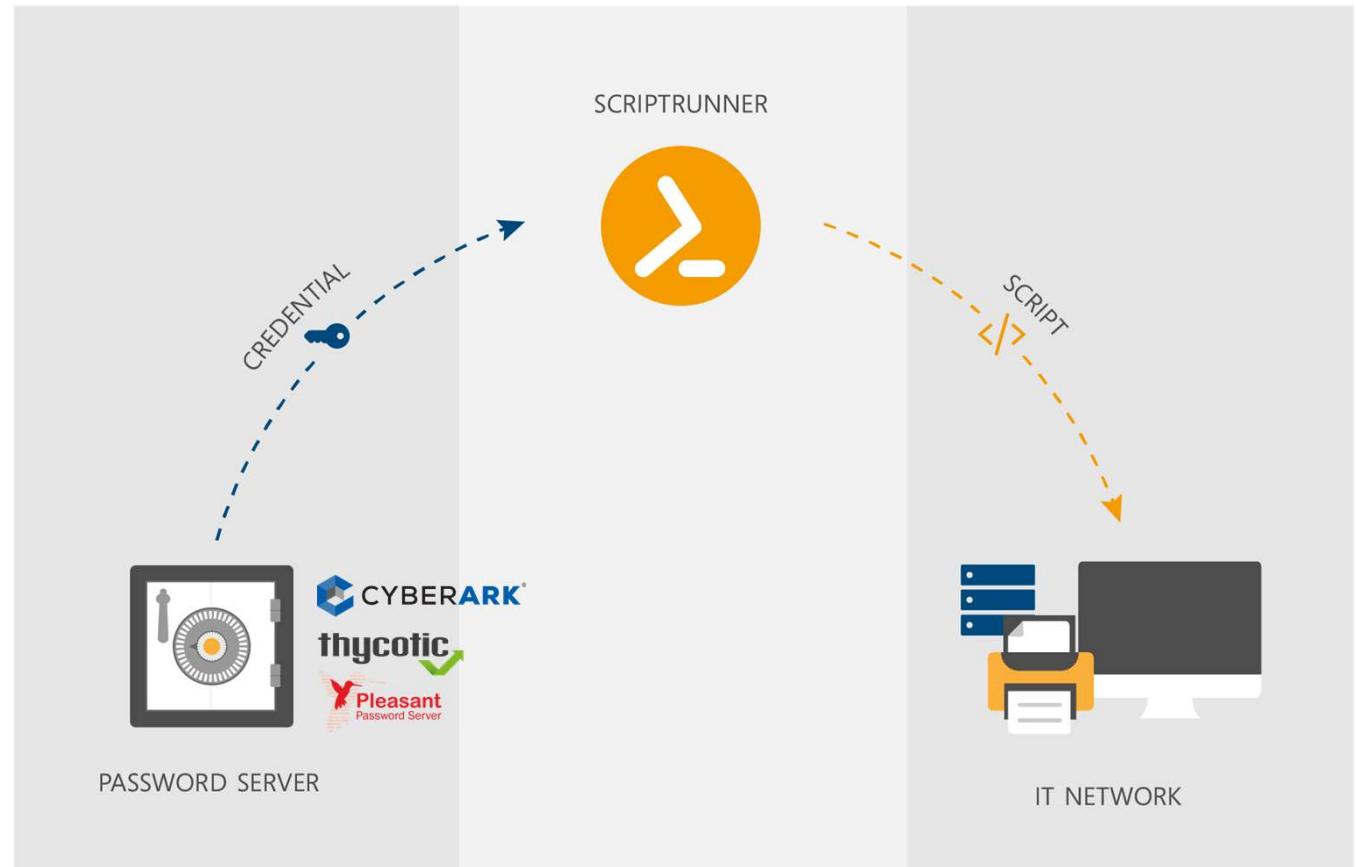
github.com/scriptrunner/ActionPacks



Password Server Support

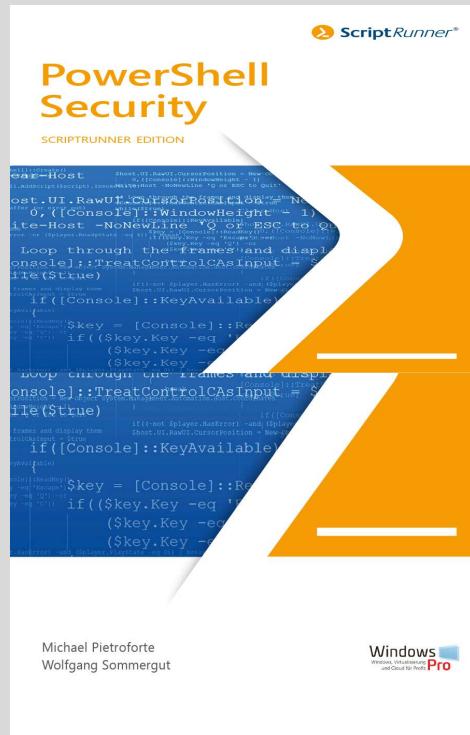


- Centralized password safes
 - CyberArk Password Vault
 - Pleasant Password Server
 - Thycotic Secret Server
- No credentials are stored locally
- Automatic password rotation
- One single secure password repository for multiple ScriptRunner instances



Free ebook: PowerShell Security

lp.scriptrunner.com/en/powershell-security-guide



PowerShell Security

Table of contents

1	PowerShell as a hacking tool: Prevent abuse of scripts	8
1.1	Lax default configuration of PowerShell	9
1.2	Hacking tools for PowerShell	10
1.3	General blocking of PowerShell	12
1.4	Circumvention through alternative shells	14
1.5	Secure PowerShell with integrated mechanisms	15
2	Restrict execution of scripts	20
2.1	Setting an execution policy	20
2.2	Siging PowerShell scripts	25
2.3	Reduce PowerShell risks with Constrained Language Mode ..	36
3	Secure communication	48
3.1	Installing OpenSSH on Windows 10 and Server 2019	48
3.2	PowerShell remoting with SSH public key authentication	57
3.3	Creating a self-signed certificate	64
3.4	Remoting over HTTPS with a self-signed certificate	71
4	Just Enough Administration	81
4.1	JEA Session Configuration	81
4.2	Defining and assigning role functions	92
5	Audit PowerShell activities	98
5.1	Log commands in a transcription file	98
5.2	Scriptblock logging: Record commands in the event log	106

PowerApps/Automate & ScriptRunner Scenarios



Automate & Delegate Recurring
PowerApps & Power Automate
Tasks with ScriptRunner



Trigger Script-executions in ScriptRunner
through
PowerApps & Power Automate



Power Platform and ScriptRunner



Trigger ScriptRunner Actions with a Power Automate Flow

Custom Connector enables ScriptRunner Actions

Custom Connector communicates with MS On-premises data gateway

SR Web Service Connector receives REST call and triggers SR Action

Create Swagger file with SR Actions in SR Admin App

A screenshot of the ScriptRunner Admin App interface. At the top, there is a list of connectors:

- Matrix42 Workspace Management
- Nagios / Icinga Monitoring
- PRTG Monitoring
- SAP
- ServiceNow ITSM
- Test WebService

The "Test WebService" connector is highlighted with a yellow background. Below the connector list is a dark grey footer bar containing four icons: a trash can labeled "Delete", a checkmark labeled "On", a power button labeled "Off", and an export icon labeled "Export OpenAPI".

Demo

Manage PowerApps & Automate with ScriptRunner



ScriptRunner – How to run scripts



Manually

Users execute tasks in web browser



Scheduled

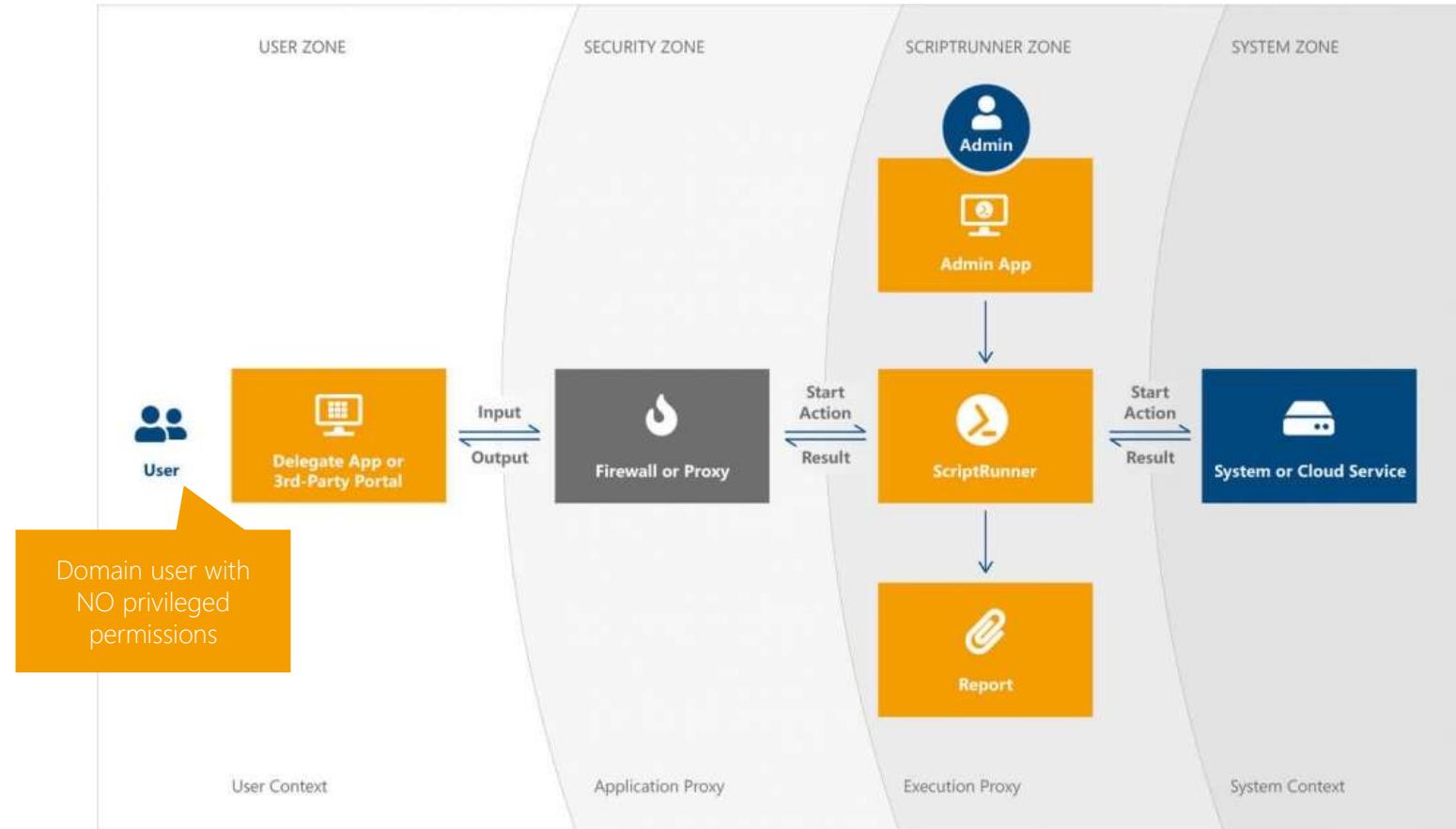
Time-controlled tasks



Automatically

Event triggered by 3rd party systems

ScriptRunner – Secure Delegation



<https://www.scriptrunner.com/de/software/system/>

ScriptRunner test report fom a Microsoft MVP



Adam Bertram aka Adam The Automator
Microsoft MVP Cloud and Datacenter Management

"ScriptRunner is a master of organizing, categorizing and delegating scripts."

"ScriptRunner is a real DevOps tool for PowerShell scripters."

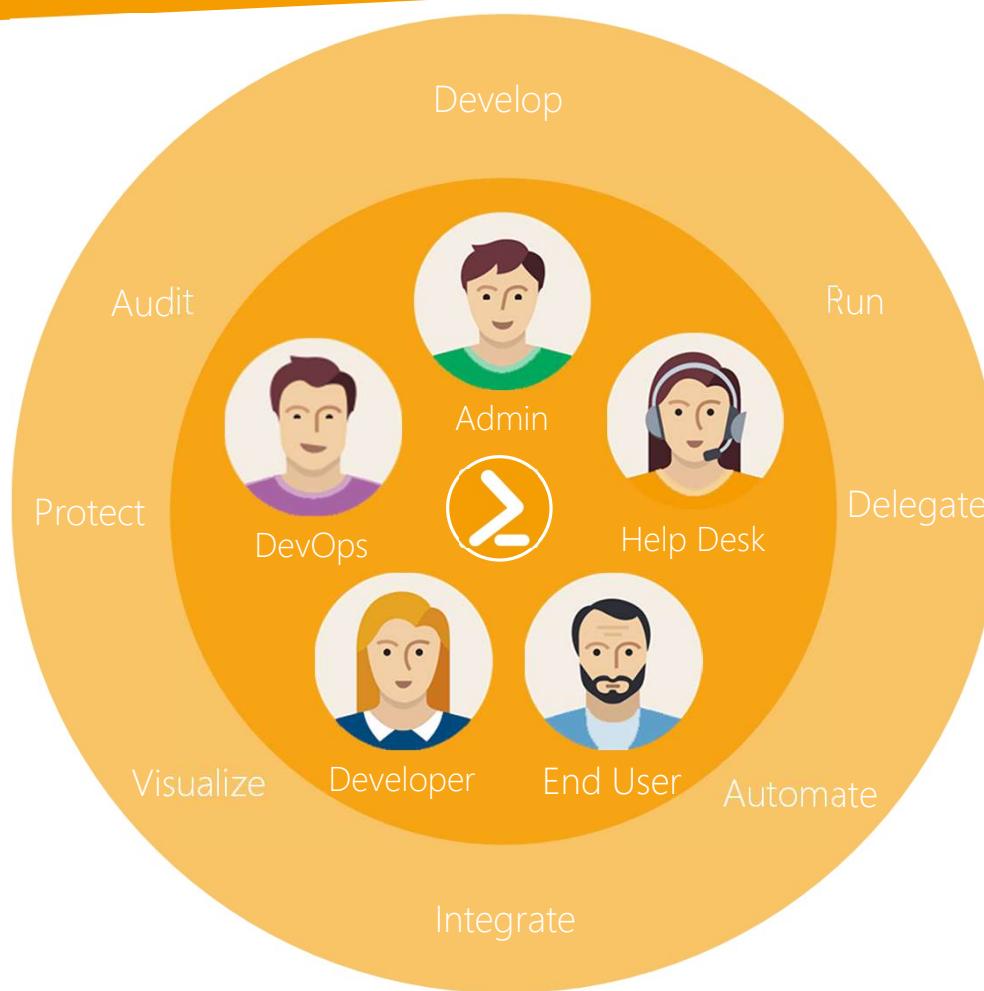
"Compared to large automation platforms, ScriptRunner is significantly cheaper"

"to get the same functionality of ScriptRunner, it would require several applications to get a similar feature set"

"If you are looking into automating your administration and support processes easily and securely, you will find a powerful and professional tool here."

<https://www.adamtheautomator.com/scriptrunner-bringing-powershell-to-devops/>

ScriptRunner – Your Smart PowerShell Control Center



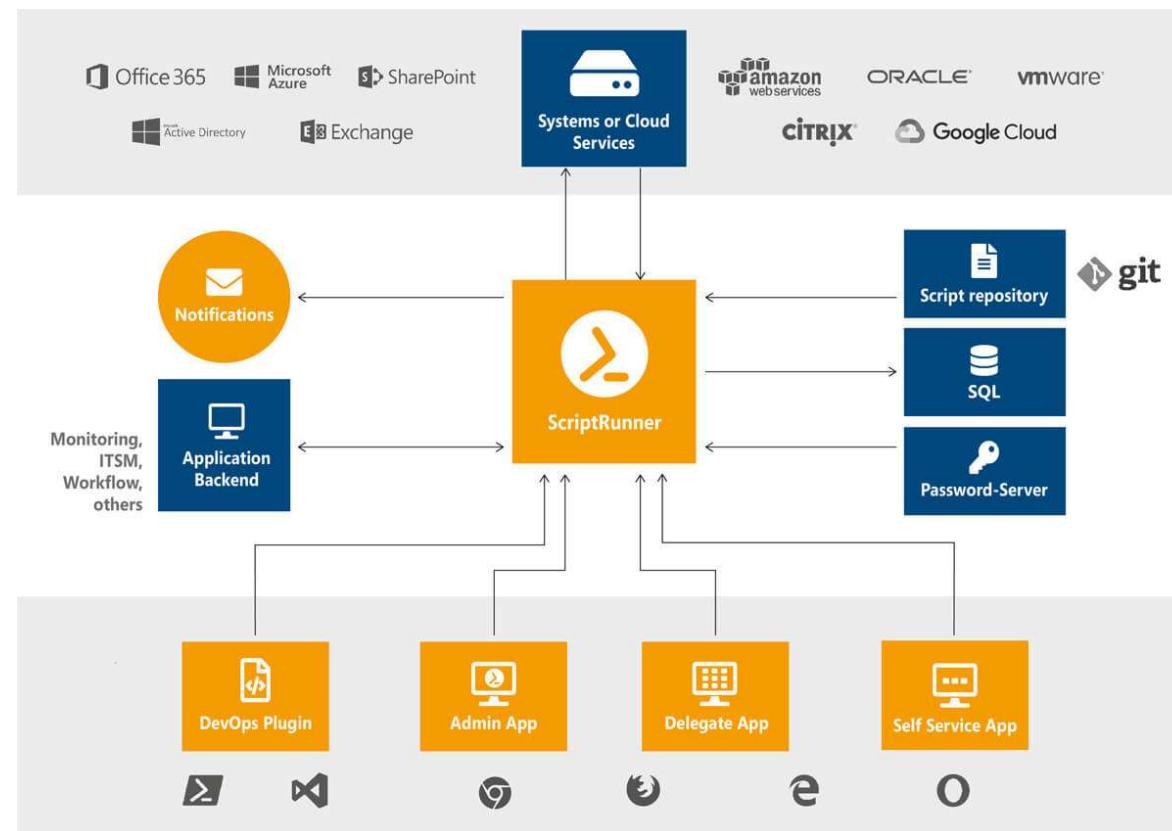
ScriptRunner – Make PowerShell A Real Solution



The image displays a user interface mockup for ScriptRunner, organized into several horizontal sections:

- User Roles:** A row of five circular icons representing different user profiles: Admin (green shirt), DevOps (purple shirt), Developer (blue shirt), Help Desk (yellow shirt with headset), and End User (blue shirt).
- Icons:** A row of five white icons on an orange background: a double-headed arrow, a padlock, a folder, three people, and three interlocking gears.
- ScriptRunner Logo:** The ScriptRunner logo, consisting of a white 'S' icon inside a circle followed by the text "ScriptRunner®", centered on the orange background.
- PowerShell Prompt:** A large white '>' symbol inside a blue square, representing the PowerShell command-line interface.
- Supported Platforms:** A row of various technology logos on a blue background, including Windows, Docker, Jenkins, SharePoint, Azure, Citrix, Oracle, Oracle Database, Docker, AWS, and others.

ScriptRunner – How the components play together



<https://www.scriptrunner.com/en/software/>

ScriptDeveloper – DevOps Plugin



- ISE Plugin
- Team-based script development
- Central script repository
- Filters
- Check-in with history
- Check-out
- GitHub integration
- TFS integration
- More information
 - <https://www.scriptrunner.com/en/software/scripts/>

The screenshot shows the Windows PowerShell ISE and the ScriptRunner interface. The PowerShell ISE window displays a script named '_UPLOAD_StartServiceChoice.ps1' with the following content:

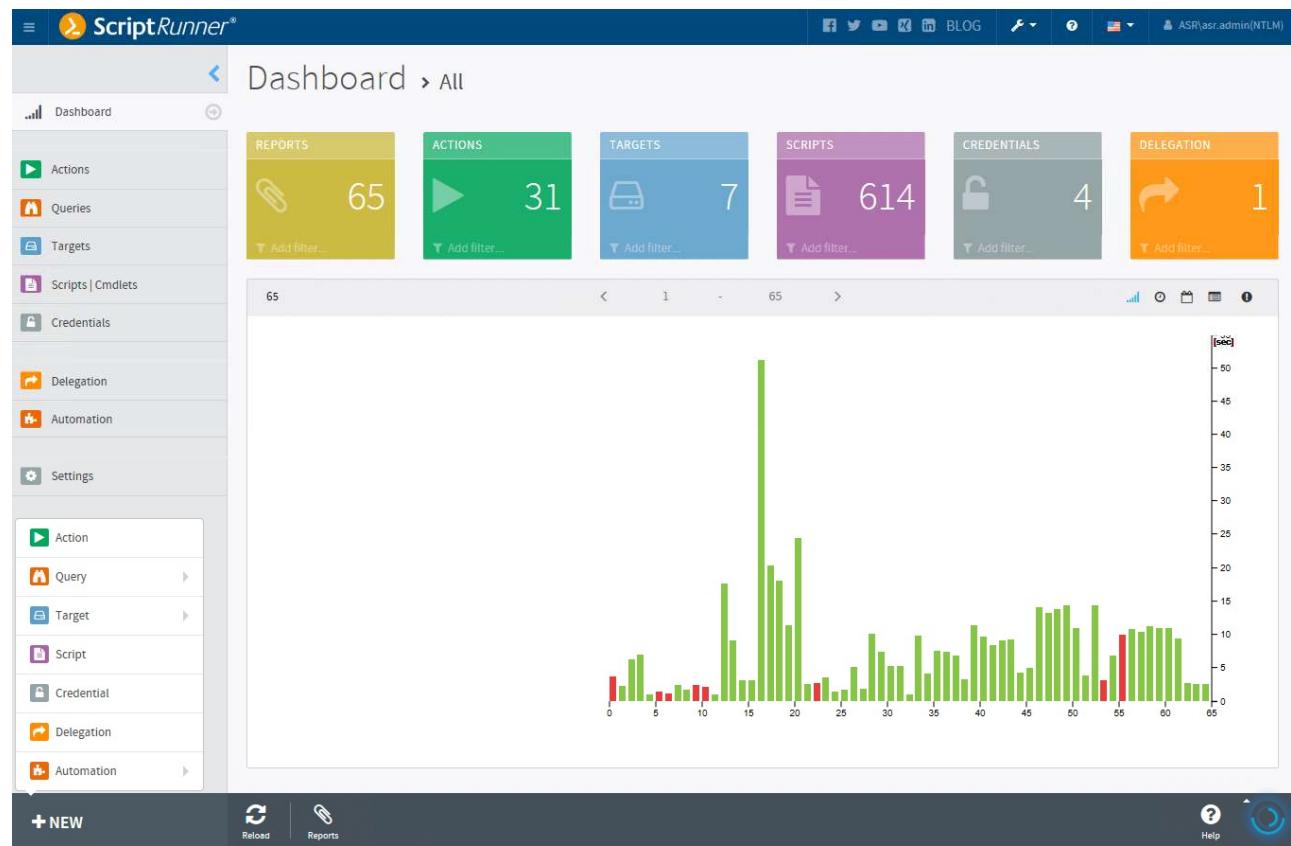
```
1 #ASR Parameter:ServiceDisplayName Type:Choice(ServicesList)
2
3 <#
4   .SYNOPSIS
5   Start a Windows service from the list.
6
7   .PARAMETER ServiceDisplayName
8   The DisplayName of the service to start.
9 #>
10
11 Param (
12   [Parameter(Mandatory=$true)]
13   [string]$ServiceDisplayName
14 )
15
16 $name = $ServiceDisplayName.Substring(2)
17 Write-Host "Starting Windows service '$name'..."
18 Start-Service -DisplayName $name -PassThru
19
```

The ScriptRunner interface on the right shows a list of available scripts, with 'StartServiceChoice.ps1' highlighted.

SysAdmin – Admin App



- A web GUI for PowerShell
- Dashboard for all activities
- Create ScriptRunner actions
- Manage and test queries
- Target configuration
 - Direct local
 - PowerShell remoting
 - Implicit remoting
- Secure delegation of ScriptRunner actions
- Scheduled script execution
- Secure credential handling
- Configure automation connectors
- Organizing by tags
- Reports and Logging
- More information
 - <https://www.scriptrunner.com/en/software/web-apps>



ServiceDesk – DelegateApp



- A web GUI for PowerShell
- Execute PowerShell scripts in the browser
- Delegate specific ScriptRunner actions to team members, helpdesk or endusers
- No administrative permissions for users required
- Parameters for script execution can be hidden
- Input dialogs are automatically created from PowerShell script
- Script execution logging and reporting
- More information
 - <https://www.scriptrunner.com/en/software/web-apps>

A screenshot of the ScriptRunner DelegateApp interface. The top navigation bar includes tabs for Favorites, All, ActionPacks, ActiveDirectory, BitLocker, Common, and Computers. On the right side, there are buttons for adding a new item, a search bar, and user authentication (CUST:cst.admin(NTLM)). Below the navigation is a grid of 16 colored cards, each representing a different PowerShell action or task. The cards are arranged in four rows of four. The visible titles include: Add-MsUsersToRoles, Add-PrtMPrinterConnection, Clear-SysMEventLog, Copy-ADUser, dbgecho, Disk Info, Enable-SysMBitLocker, Enable-SysMLocalUser, Format-FillVolume, Get-ADComputersWithDefinedStatus, Get-ADUserProperties, Get-ExMailboxProperties, Get-SnapshotPropertiesFromHyper-V, Get-SnapshotPropertiesFromVMware, Invoke-GitSync, and Local: Add two values. Each card has a star icon and a small edit icon in the bottom right corner.

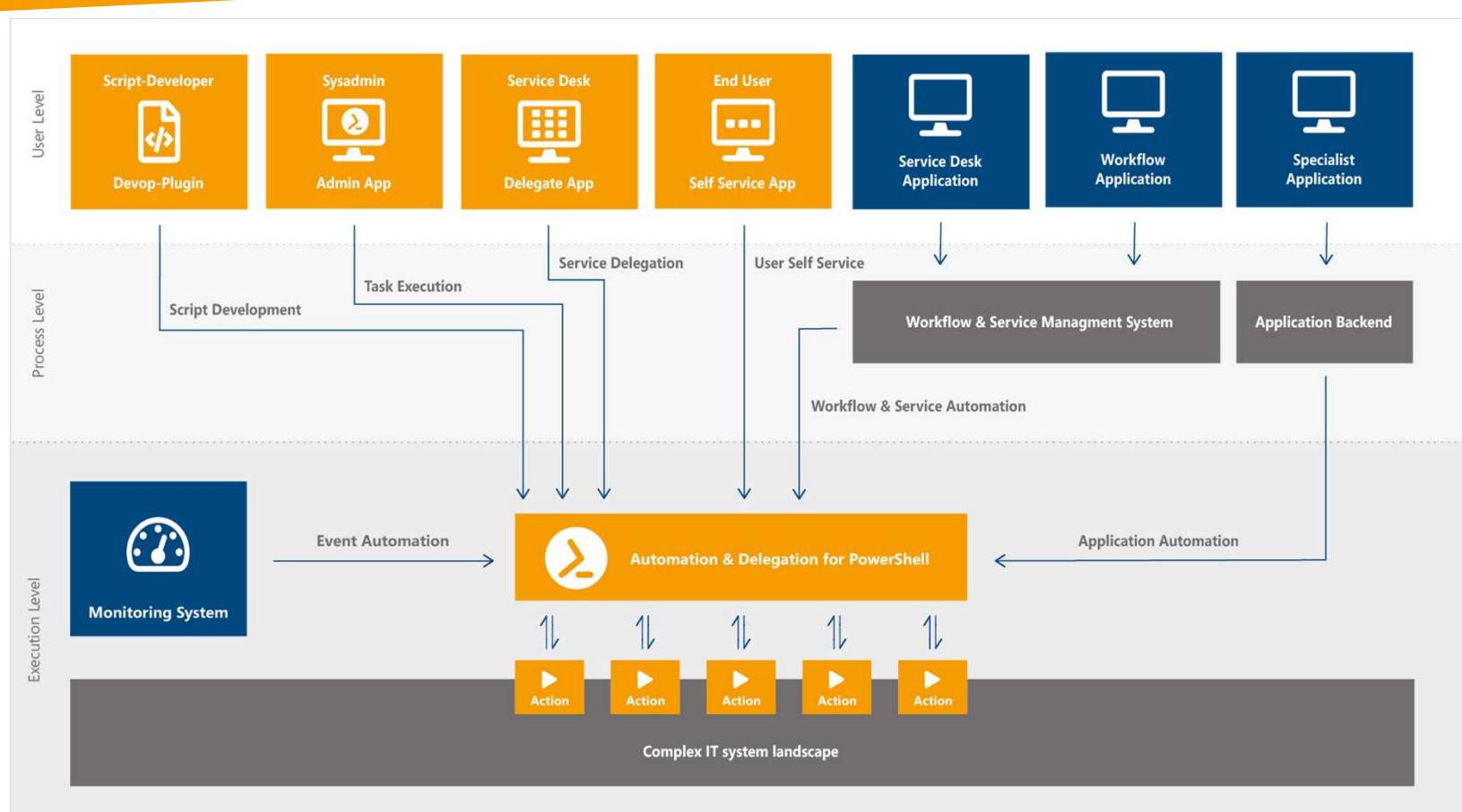
Integration – Add new functions with connectors



- 3rd party systems automatically trigger ScriptRunner Actions
 - Monitoring Systems
 - IT Service Management
 - Business Workflow
 - Applications
- Password server integration
 - CyberArk
 - Pleasant
 - Thycotic
- More information
 - <https://www.scriptrunner.com/en/software/automation-connector/>



ScriptRunner – The Full Picture



<https://www.scriptrunner.com/en/software/system/>

Unlock the Power of PowerShell across your organization



Manager

Our organization becomes more productive

Deliver great scripts in a secure environment



DevOps

Delegate recurring tasks and save time



Administrator



Developer

Develop scripts for more use cases/users



HelpDesk

Deliver more & better services in less time



EndUser

Faster task completion with self-services

Put ScriptRunner
to the test.
30 days for free.

lp.scriptrunner.com/en/demo-download



Let's talk about your
automation use cases.
Book a online session
with me.

[lp.scriptrunner.com/meetings/heiko-
brenn/demo-en](https://lp.scriptrunner.com/meetings/heiko-brenn/demo-en)



Put
ScriptRunner
to the test.
30 days for
free.

lp.scriptrunner.com/en/demo-download

www.scriptrunner.com



Ludwig-Erhard-Straße 2 | 76275 Ettlingen | Germany

Tel: +49 7243 20715-0
Mail: info@scriptrunner.com