

VERSATILE HELPER

# PowerShell in IT-Support

LOOKING BACK:  
2 YEARS AFTER PUBLICATION IN  
IT-ADMINISTRATOR

BY CLEMENS FEIGL

---

*When I was invited to publish the following article in IT Administrator in 2018, many IT teams were facing challenges of 2nd and 3rd level overload. Reacting instead of acting - the same picture everywhere. In the following, I have reflected on the last two years through many discussions and projects with our customers and comment on the core question for each paragraph: Will it still look the same in an automation year 2020 or what has changed the situation?*

---

**2018:** The increasing complexity of IT operations is bringing IT departments to the brink of exhaustion: Routine tasks in particular cost time, which in turn leads to less innovation and less drive for new and exciting projects. However, it is precisely these that bring companies forward in the age of digitalisation and working 4.0. The PowerShell can help admins in IT support and take over tasks.

**2020:** These effects have become even stronger as more and more services are moved into the cloud and even more complex hybrid structures are in operation. Work 4.0 is now called "New Work", but many customers confirm that work 2.0 is still going on.



With today's demands on IT departments to drive digitization, unnecessary and repetitive support tasks are a headache and hard to cope with. One of the main reasons for this is the lack of automation and delegation of processes to the helpdesk or support department. Optimally coordinated IT teams as well as the right processes and solutions would be necessary. As a result, increasingly complex IT landscapes, supplemented by cloud services, can no longer be efficiently administered by helpdesk staff.

**2020:** In this respect everything is still based on the above point. In addition, 2nd / 3rd level IT staff are often frustrated when they have to correct mistakes resulting from time pressure and lack of concentration (especially when working on annoying tasks).

## EFFICIENCY FOR TEAMS AND CUSTOMERS

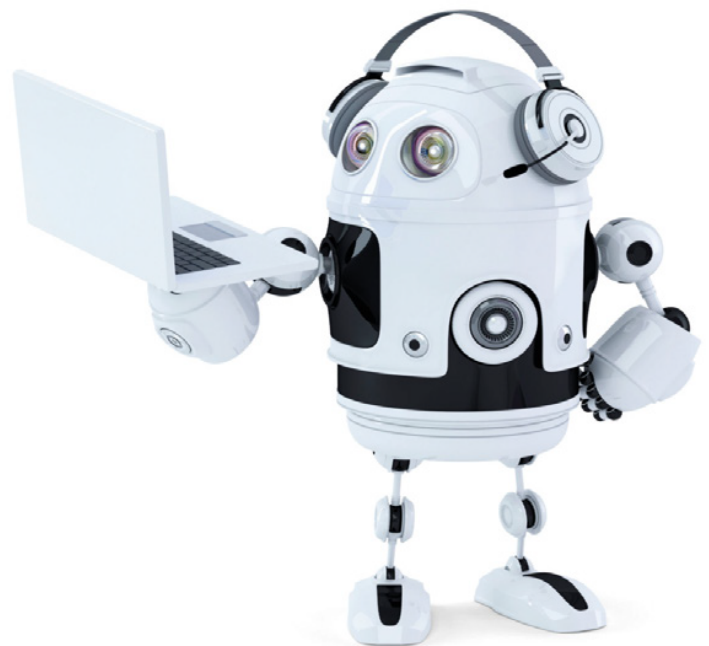
Work and time-intensive tasks are commonplace in IT - but they are undesirable and unnecessary with repetitive helpdesk and support calls. The dream of IT operations: a quick and easy automation of administrative processes and routine activities. And: a secure delegation of tasks to helpdesk, support and other teams or departments. Customers and users also demand shorter reaction times, which are often also stipulated in the SLAs.

**2020:** More and more often the support desk acts as a stopgap, the colleagues there don't feel comfortable with this role and would like to have a better internal image in addition to the entire IT team.

So what is stopping companies with this knowledge from implementing a solution that supports the IT team and allows innovation? The cost side is often cited as the reason. Many platforms for automating and delegating IT services are expensive and require an exorbitant implementation effort - apart from the loss of control feared by companies and the worry of no longer being able to meet compliance requirements with automation. Furthermore, these logic layer orchestration platforms are very complex, and the creation and design of workflows or runbooks is often the responsibility of very small teams. These are primarily concerned with IT process design or the development of IT frameworks.

In IT operations, however, we are at the technical implementation level. Here we are dealing with system-specific requirements that are implemented by specialized teams or administrators. Use cases, for example in the Active Directory or Exchange environment for delegation or automation, are implemented faster and better in these specialized teams than at the logic level of the orchestration platforms. This means that automation and delegation can also be used directly. Long adjustment cycles are no longer necessary, because reproducibility, efficiency and risk avoidance are essential.

**2020:** Decisions made at the process level still result in the introduction of tools that are unsuitable for the actual application (e.g. too simple or too complex). This makes work even more time-consuming and frustrating (e.g. due to lack of usability or long project durations).



## POWERSHELL PROVIDES REMEDY

PowerShell can be used to mitigate any arguments against automation and delegation. It allows high complexity to be reduced through modularization, and processes become highly reproducible. The PowerShell is available for many hardware and software solutions, enables simple automation even in complex landscapes and supports Windows, Linux and Mac OS. This means that the script language can be used very broadly as a cross-sectional technology, whether for Windows servers and clients, Linux, Mac OS, Microsoft Exchange, Microsoft SharePoint or even cloud variants such as Office 365. Third-party systems and applications range from VMware to Citrix and IBM, which now offer a wide range of cmdlets that can be used to standardize various smaller tasks.

**2020:** This will be more true in 2020 than ever before. More and more modules for solutions such as Microsoft Teams or Power Apps etc., up to PowerShell 7 (cross-platform) can be used and are constantly improved and extended. Even PowerShell reporting with personalizable HTML-based reports can be configured.

The main problems here are, as so often, security and usability. On the one hand, the user needs administrative rights to be able to log on to the target system to execute the scripts, and on the other hand, PowerShell scripts contain administrative credentials of the

Target systems that are not intended for all IT staff. This also affects traceability: if the employee quits the PowerShell console after execution, the data is practically gone. It is no longer possible to trace who executed which script on which system with which rights and which account. If logging mechanisms of the latest PowerShell version are used, exactly the opposite happens: Admins almost drown in the abundance of non-expressive individual data. To make matters worse, working with the PowerShell console is not user-friendly and often causes an even higher ticket volume due to misuse and lack of skills.

**2020:** This problem also exists with "homemade" PowerShell frameworks. In addition, these are not standardized tools and they are often dependent on a single person - the creator.

## ALL UNDER THE SAME ROOF

IT departments therefore have numerous requirements, which at best should all be fulfilled by the PowerShell simultaneously. These include leeway through the delegation of routine activities, transparency for reporting and auditing, the development of new opportunities through automation and delegation - and low implementation costs. This wish is not unrealistic.

In such a scenario, all activities and tasks in IT operations are ideally automated and delegated. DevOps and administrators create scripts using an ISE and store them in a central repository. The administrators then define the appropriate execution policies for the scripts in their respective subject areas.

**2020:** The choice of tools has changed: instead of the ISE discontinued by Microsoft, the focus is now on Visual Studio code or similar. Also professional code management systems like Github, Gitlab or similar are used.

On this basis, individual tasks and activities can be delegated both between IT teams and to sub-administrators, helpdesk and support, as there is a logical separation between the target system and the credential, which are only merged in the background during execution. Selected execution policies can be made available to end users, for example by integrating them into an existing portal (service portals, etc.) or via a web interface that allows operation without additional developer know-how.

**2020:** It's a lot safer here today. Password servers like Thycotic, CyberArc or Pleasant are becoming more and more popular. The service desk user triggers an execution via the WebGUI, the central component gets the required credential from the password server and executes the script centrally and securely.

For automation purposes, monitoring and ticket systems as well as other applications for service management are linked via connectors, ideally via WebService interfaces. In addition to the delegation of scripted, operative routine tasks and time-controlled processes via script, event-driven automation with PowerShell has considerable potential. And not only in efficiency, but also in quality, reproducibility and traceability.

## AUTOMATE MONITORING

Monitoring systems, for example, are being equipped with increasingly intelligent sensors and recognition algorithms to detect system errors and statuses. However, the counterparts required for extensive automation are missing. With PowerShell scripts, these intelligent actuators can be created for any individual system environment. However, automation platforms can also be used to ensure the management of the script actuators and their execution in the respective context and to meet safety requirements. In the event of an error, the monitoring system would trigger a suitable actuator, which would then process the sensor information in a safe administrative context and automatically correct the problem.

**2020:** Here are some suggestions for further possibilities: Even microservices can trigger fully automatically via REST on an event-driven basis. The possibilities have become almost limitless.

In order to be able to integrate other legacy systems or applications without web service technology into an automation, further mechanisms are required. An example of this is the sending of e-mails using templates and variables or parameters. These system e-mails can also be used for automation by starting scripts for automation via e-mail using appropriate execution guidelines.

**2020:** These systems can still be found in customer environments, although it is clear that REST is the tool of choice.

Where otherwise several IT solutions had to be used in parallel to get all the wishes under one umbrella, this approach can now be used by ScriptRunner, for example. The platform combines script development with automation and delegation as well as execution and monitoring. Thus, tasks within and across IT teams can be automated and delegated. These tasks can then be delegated to helpdesk, support or the user for policy-based and secure execution.

**2020:** The ScriptRunner PowerShell Management Platform now serves even end users via the Self-Service App

Integration with third-party systems allows complete automation including automated documentation of completed tasks. Even in the case of an audit, companies have all the evidence on their side, as detailed reports are recorded and, if required, stored externally on a long-term basis. Each of these steps noticeably reduces the reaction times and the effort per task, the reproducibility and thus the security increases considerably, the efficiency in the team increases.

**2020:** If one believes studies that have tested automation in the Ops environment, companies will have up to 10 automation tools in use, which should be able to interact with each other.

## CONCLUSION

There is a demand for automation in many areas. The approach for interaction with third-party systems provides for automatic reaction to status changes or the reaching of threshold values on target systems. Anyone who wants to guarantee users a reliable handling of service requests and IT staff the freedom to develop new ideas and consistently implement digital transformation should act quickly. No company can afford repetitive, manual and error-prone tasks anymore.

**2020:** No digitalization without automation!

*Clemens Feigl is Head of Product Sales at ScriptRunner Software GmbH.*