S taying afloat is difficult enough for any IT administrator. For many, the root problem is that every issue is individually addressed. Problems (and their solutions) should be linked to a broader workflow. To get ahead, administrators must connect independent solutions together. These solutions allow entire business processes to be automated and connected.

In this essential guide, we will cover the three big problems that every IT shop faces. While each problem will seem like a separate piece, solving one enables you to tackle the next and to tie your solutions together. The time saved by automating the first problem compounds into viable solutions for the second – and so forth. Before exploring the problems, we will examine the different levels of tools that we can use in each task. Each level represents an evolution on the previous toolset. Seen together, we have a clear picture of where process management is headed. This picture will also show why many solutions end up as a kludgy process and how the entire solution lifecycle can be streamlined.

Our first core problem is object automation. When it isn’t in place, organizations waste time and money waiting on IT to manually create and change accounts.

Any person who has waited for IT to create their email or reset their password knows how frustrating and slow this can be. Solving this problem allows employees to access what they need immediately and frees up IT administrators for more impactful projects. This section will focus on user account automation across platforms.

The second problem will dive into content management and data sprawl. While discussing this common issue, we will also explore the impact that each level of tool has upon our solution. These tools are applicable for environments running traditional file servers or storing data in external applications. Problems and their solutions can involve platforms ranging from Amazon Web Services to Microsoft Azure. Solving this problem will allow your organization to use limited resources more efficiently. A full solution can provide insight on how data is being generated, where to appropriate additional assets, and where content can be stored.

In our final section, we will discuss notification and data synchronization. Specifically, we will see how adding these two features to any process is a must for business continuity. Notification becomes extremely necessary when automated processes are used. A breakdown here will affect a larger section of the organization. Data synchronization and manipulation is required when applications do not easily talk with each other. Ensuring that each process implemented has detailed logging and notification abilities will allow for administrators to track what changes. Many solutions lack the ability to synchronize and manipulate data. Being able to accomplish both of these in a streamlined fashion removes one of the last manual intervention steps many organizations deal with.

THE THREE LEVELS OF MANAGEMENT TOOLS

Each solution requires a variety of people and tools to succeed. There is a hierarchy to this automation and to the tool that you choose to utilize. Choosing the wrong type of tool (or wrong level of tool) will add countless extra hours of work.

What exactly do we mean by level of management tools?
At the most basic level, an administrator uses a single dedicated tool to tackle repetitive problems. Each tool has its own unique GUI and terminology that must be mastered before the tool can be useful. A set of tools may belong to the same suite and still be at this basic level. In fact, it is common for management software in the same suite to share nothing more than a common name. In this guide, we will call these basic management applications level-one tools.

The evolved version of the dedicated management tool is a shared scripting language or platform. In the world of Windows, PowerShell is the prime example of this. With modules, which each product team designs, a common scripted method of management emerges. This level of management carries many advantages over single use tools. Common parameters allow knowledge to be reused and object editing makes data manipulation easier. On the flip-side, scripted management lacks the inherent ease of GUI management. Because these solutions are developed by single enterprises, automation is normally limited to their scope of products. This had led to the concept of devop administrators (administrators who write and develop their own tools). Unfortunately, this method of management is entirely too specialized and time intensive. This guide will refer to these applications as level-two tools.

The fully evolved management tool is independent of a specific vendor. Instead, it utilizes a variety of management frameworks under a single automating application. This allows processes to talk across ecosystems by using common formats (see the Notification and Syncing section for more information on this). The more advanced this tool is, the simpler the automation process becomes. Though this seems counterintuitive, solving a complex process should not be made harder through random syntax and single use knowledge. Eventually, these management tools connect intricate background processes with straightforward and streamlined GUIs.
example, a 300-line level-two script can be reduced to a simple drag–and–drop action. These solutions will be denoted as evolved or level-three tools.

With a thorough understanding of the toolsets available, let us explore our first big problem—object and account creation.

**CONTROLLING OBJECT CREATION**

IT is relied on for a lot. Object creation—specifically user accounts—is one of the most important and time sensitive processes handled. Controlling user creations can involve quite a bit of repetitive work. This is especially true if data is not centralized in at least one location. Every environment is different, and this section will provide a high-level look at the problems organizations face and the multiple paths to tackle them.

What makes object creation so difficult is not as much technical as it is logistical. The first hurdle for an organization to overcome is identifying what will be automated. Will every account type be automatically created or just the necessary few? If the accounts are created automatically, will passwords (and other data) be synced as well?

These questions bring us to the second hurdle of tool selection and to the beginning of this automation. When an IT department chooses to utilize a level-one tool, an administrator may end up using a dozen different applications just to create user accounts! The effort to learn each application and the huge potential for human mistakes show why this second hurdle is often more difficult than the first.

To illustrate this point, we are going to look at three ways an organization could create users in Active Directory.

**HOW OBJECT CREATION DIFFERS WITH THE TOOL YOU CHOOSE**

Many tools exist to provision objects in Active Directory. They cover the spectrum on complexity and ease of use. Each tool fits into a distinct level.

At most, a level-one tool would allow limited automation. Notable examples are the built-in management GUI tools: Active Directory Users and Computers (ADUC) and Active Directory Administrative Center (ADAC). An administrator using a tool like ADUC or ADAC can only create objects as fast as he can type. Data input is limited by him and progress cannot occur exponentially. In fact, the entire system relies on the IT department doing a manual process that is prone to mistakes (misnamed accounts, incorrectly entered attributes, insecure settings, etc.). This is a prime example of why level-one tools hold departments back.

Level-two tools can automate more but still suffer from some of the drawbacks mentioned above. The biggest issue with a level-two tool is the cost of simplicity. An example of a level-two tool would be a user account creation script. Using this tool either requires specialized knowledge and usually some trust in an external resource (such as an online code repository). Most level-two resources are borrowed rather than created in-house.

Level-two tools involving custom scripts have several shared problems:

1. How easily is the solution customized to run in your environment?
2. If the tool isn’t developed in-house, how much trust do you have in an online free resource?
3. Where is logging done? How will an administrator know if an account is not created? What is done to rectify failures? Many scripts do not provide that functionality natively.

Employees will likely need more than a single username, as your environment will likely have many applications. Some will integrate with Active Directory or support a single sign-on mechanism. Others will maintain their own username/password database. Level-two tools are not adequately suited for the task of integration; they simply are not flexible enough.

A level-three tool is able to work around these limitations. Because this solution type includes the framework to automate across platforms, a department doesn’t have to piece together multiple level-one or level-two tools. A level-three tool can tie into the entire business workflow. A fully evolved tool can easily gather inputs from third party applications, pass these to the account creation process (regardless of underlying infrastructure), create the objects, and report on failures/successful created objects. Each action is reportable.

REAL BENEFITS FOR OBJECT AUTOMATION

Tackling this first problem should be a priority for every IT department. The issue is too important to leave as a manual process. An organization with a fully automated object creation structure enjoys enormous advantages. Newly hired employees are able to quickly integrate themselves instead of waiting on their accounts. New systems can maintain the same username/password structure of existing authentication resources. This simplifies the day-to-day tasks of departments. Finally, the IT department is able to extract themselves from the repetitive work of object maintenance.

MANAGING FILES, SHARES, AND APPLICATIONS

An object by itself isn’t worth very much though. It is the generated content that provides direct value to an organization. Users will create data and that data has to be managed. In this section, we will discuss the problems associated with managing on premise and externally accessed data. These problems are made more complex when an inadequate tool is chosen for management purposes. This section will start with traditional data access and expand information stored within third party applications.

Like object automation, data management problems can be divided into technical issues and organizational issues. Without a complete understanding of the organizational portion, a true technical solution will remain elusive. Getting a handle on this content creation requires a top-down approach to structure
and organization. Questions to ask yourself include:

- Where exactly will content be stored? Local site or central location?
- Who manages content security and enforces organizational policies?
- How will server resources be managed and shared?
- How will shares be managed for different departments or sectors?
- What tools will be utilized for a technical solution? How well will they integrate?

With these questions answered, you can begin pursuing a technical solution.

**CHOOSING A TOOL FOR APPLICATION AND DATA MANAGEMENT**

As more applications are used in an environment, control ebbs between IT and the application itself. When an environment has to deal with so many diverse programs, support is made more difficult because every application has its own unique method of interaction.

When an application does not provide native automation support (which many don’t), the organization has three real options:

4. Provide manual integration. This is normally accomplished by long repetitive work and is often delegated to the IT department.

5. Purchase another similar application that provides integration features.

6. Pursue an automating application suite.

All three choices have real costs to the organization. The first choice trades the time saved in integration by the cost of inefficient labor. This is our level-one solution. The second choice presents an opportunity cost because of the application change along with the true cost of purchasing two sets of software. If the organization purchases another single use application, work is simply repeated. A level-two solution may be pursued here, but the cost of time can skyrocket. The third choice adds an additional layer to application management, but is the best long-term solution. It provides a clear way to tie in future applications without the time cost of learning single use skills. This level-three application abstracts itself from the problems that tie the first two options down.

**NOTIFICATION AND SYNCING**

A good tool needs the ability to provide notifications/logging and synchronization across a variety of platforms. When your choice of tools doesn’t include either feature, you must manually check processes for completion. This intervention completely defeats the point of automation.

Notifications can be handled within the application itself, but it is far better if the notifications can be centralized. Two common methods for notification are a central syslog server and good old-fashioned email alerts. A syslog server and email alerts should not be the primary method of logging though. Think of these alerts as threshold notifications. When desired, granular logging should be included (or easily added) to any process.

Synchronization across a platform is how you take your final data and get it to the recipient. Often, FTP or email is used to accomplish this. Many scripting platforms lack easy ways to handle FTP. Code is borrowed from public repositories and can be difficult to understand. Email, provided the attachment is small enough, can be handled in most environments. Let’s look at how these two features differ across the three levels of tools.
HOW NOTIFICATION AND SYNCING TIE INTO THE THREE TOOL LEVELS

A level-one tool may or may not have notifications and syncing built into it. If it lacks these features, another tool must be set up to monitor processes for completion or errors. Any information that must be transferred to another application will probably need to be processed manually.

A level-two tool may be able to include email alerts easily, but will probably lack advanced filtering that a level-three tool includes. The ability to control the severity of the alert and the notification method is an added bonus to this functionality. Data manipulation is easier than a level-one tool, but getting the correct formatting can still present challenges. Because level-two solutions are syntax oriented, simple mistypes can prevent alerts from reaching their targets or cause data loss.

Level-three tools have robust notification features and the ability to catch/resolve any potential errors. In fact, a process utilizing a level-three tool should be able to follow any workflow that is generated for it. Level-three tools are data agonistic. They do not care where the data is or where it is going. This allows for extensive data manipulation and syncing.

Adding notification and syncing features to critical processes should be a priority for any IT department. Automating this allows for a proactive response to task breakdowns, eliminates the need to manually format data, and keeps everyone informed of business workflows.

INTEGRATING IT INTO THE BUSINESS

Automation is a staircase. Each step is a problem to solve and a way to integrate your solutions into a business process. Each step up represents a problem solved and an eliminated inefficiency. The tools that you choose determine how quickly your organization rises. In this essential guide, we focused on the different management tools available and how to deal with three big problems every organization has to face. These problems were object creation, managing data, and notification/syncing. In each problem, we explored how distinct levels of tools affect the solutions that are desired.

Automating each problem provides two core benefits: time saved and mistakes avoided. These benefits compound into each other. For example, accounts and objects that are automatically created require no interaction from IT administrators. This allows those employees to focus on other core issues. The processes run themselves. Automating each problem provides two core benefits: time saved and mistakes avoided. These benefits compound into each other. For example, accounts and objects that are automatically created require no interaction from IT administrators. This allows those employees to focus on other core issues. The processes run themselves.

Freeing up resources and time allow you to see (and solve) entirely new and complex issues. Streamlining these three current business processes sets the cornerstones for future solutions. Choosing the correct tool makes all of this possible.
Integrate.
Orchestrate.
AutoMate.

Manage disparate applications from a single, easy-to-use platform.

AutoMate
A Division of HelpSystems

www.helpsystems.com/automate