

# IT Administrator

*The magazine for professional system and network administration*

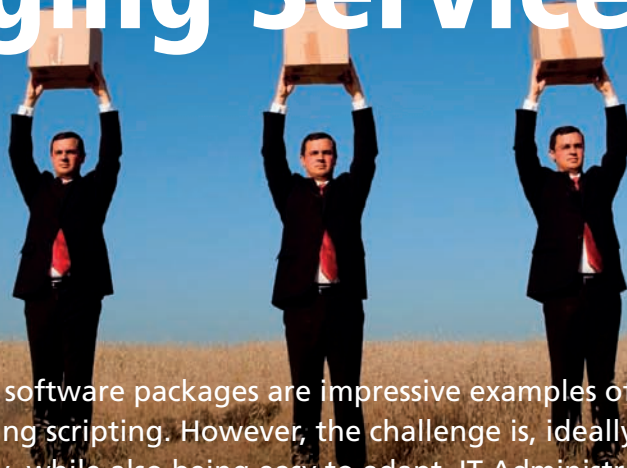
**Comparison Test of  
Script-Based Software Packagers:**

## **Competing Packaging Services**

## Comparison Test of Script-Based Software Packagers

# Competing Packaging Services

by Jürgen Heyer



Tools for the creation of software packages are impressive examples of how recurring processes can be automated by using scripting. However, the challenge is, ideally, for the scripts to generate themselves automatically, while also being easy to adapt. IT-Administrator magazine therefore decided to take a closer look at ACMP Automate, baramundi Automate, and WinCM Scripting Framework, all of which create script-based software packages. We wanted to know their scope of application and how much programming knowledge they require.

**S**oftware packaging and endpoint management are of course closely interlinked, so we had no difficulty in finding scripting tools offered by the manufacturers of endpoint management software. Although some of these modules are also offered individually, ultimately they are part of larger software deployment suites. We tested ACMP Automate by Aagon, which is also part of the ACMP Suite. The second product was baramundi Automate, which is part of baramundi Deploy. This in turn is included in the modular baramundi Management Suite. Our final test candidate was the PowerShell-based scripting framework of Windows Client Management AG (WinCM), which is specialized in software deployment, packaging, and end device/server engineering in a Windows environment.

We wanted to examine the scripting functionalities of all three tools to see how easy it is to create scripts, what programming knowledge is required, the ease with which adjustments can be made, and whether the options provided go beyond the creation of software installation packages.

### baramundi Automate

The central baramundi product is the modular baramundi Management Suite for complete endpoint management. baramundi Automate is one of the modules within this suite and is also included in the Deploy module which, as the name suggests, is used to deploy software to end devices. In doing so, the Automate module creates packages for automatic, silent installation that does not require any user input. Most software products are suitably prepared so that this can be achieved with little effort. Whenever this is not the case however, the process must be individually defined. With the Automate module, almost every process that can be executed manually on a PC can be automated.

The aim of Automate is to save the administrator from having to program by using drag & drop. The module generates response files for automatic installations and scripts for special installations and for interface controls for non-standard setup routines. Automate consists of baramundi Application Wizard and Automation Studio.

### Significant Preparations Required

Since unlike our two other test candidates, Automate is not a small, stand-alone tool that can be installed on a workstation but rather a module within a management suite, its central components are required. Therefore a baramundi Management server with Microsoft SQL Server must be installed on a server. baramundi Management Center is then the central user interface for all modules. Regarding the Automate module, it should be noted that one part, the Application Wizard, is integrated into the Management Center, whereas the other, the Automation Studio, is a separate application.

Application Wizard is the right choice for creating automatic installation packages based on most setup methods. It knows the typical installers, such as the "Nullsoft Installation System" in our test, and suggests how to proceed: as in our case entering the installation parameters using a command line. The Wizard also lists these parameters, making it very easy for the administrator to identify the one



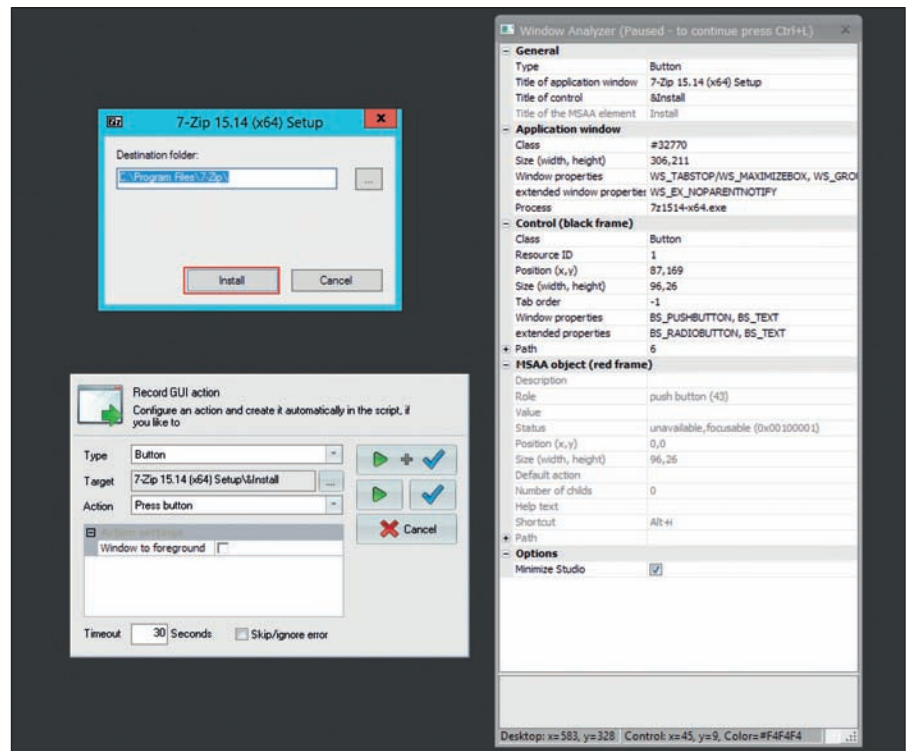
required. The result is a complete installation package that can be linked with a job in the Management Suite. If this job is assigned to an end device, the package is installed there. In general the procedure here is very convenient and consists of just a few steps.

## Drag & Drop Replaces Programming

Things get more complicated when Automation Studio needs to be used, because this is not a standard setup. Automation Studio contains two different procedures: manual scripting and GUI automation, which we describe later on.

Scripting with the Automation Suite is not about learning a programming language or writing programming code. Instead, scripting operates one level above, by defining a process as a series of actions. To do so, on the left of the GUI the user sees an actions window containing the command set. The actions are grouped by subject according to user interaction, variables, file operations, registry, user management, network functions, services, system functions, program flow, and, finally, the separate point GUI automation. Understandably, the actions themselves are based on what is required to install applications and control processes. For example, the file operations contain the actions "Copy/Delete/Move File/Directory," "Create/Delete Shortcut," "Run Variable Replacement in File," "Write Text in File," "Download File," "Unpack Archive," and "Set Value in INI File."

In order to create a script, the IT Manager drags an action into the central script window. A parameter window immediately opens. When, for example, the action "Copy File" is performed, the source and target directory must be specified here together with a filter for the file name, if necessary. The typical copying options also appear, i.e. copy write-protected files, only copy newer files, copy directories with sub-directories, copy NTFS authorizations, and create backups of overwritten files. We were particularly impressed by the copy option "Overwrite Blocked Files on Next Reboot" which, in the software installation environment, is very important for updates



The GUI recording in baramundi Automate is necessary, if all other options aren't working and actions need to be recorded within the GUI.

relating to the operating system. For the other actions, the relevant parameters are queried accordingly.

Under system functions there are some special actions such as starting and ending a process, setting the display mode, registering a component, setting the x64 mode, and installing drivers. For the latter, the INF file of the driver, on which basis the installation is performed, must be specified.

## Embedding External Scripts

If the actions of Automation Studio are not sufficient, there is an option to embed or run CMD commands or scripts (Visual Basic, PowerShell, and Java). To do so, the operator can enter the command or script within the parameter window in an edit field, paste it from the clipboard, or start the external Windows editor, import the script there, and edit it. This possibility means there are practically no limits for the Automation Suite. It also makes it very easy to integrate existing scripts.

A range of different actions which deal with structuring the script are summarized under "program flow." Branchings in the process are controlled via the action "Condi-

tion Group." Here, two operands must be entered and a comparison criterion. Alternatively, a comparison can be made based on a version number. Branchings are important if, for example, you need to make a distinction in a script according to the operating system version, in order to run the appropriate commands.

If the commands for a condition group consist of just a few actions, it is advisable to add these indented directly afterwards. Indentation via an arrow in the header of Automation Suite is important in order to indicate membership of the condition group. This is more or less equivalent to "begin .. end" or the typical "If .. endif" construction under Visual Basic. If, within a condition group, multiple actions need to be performed, it is advisable to write these in a subroutine and then to access them from the condition group. To a certain extent this is down to individual preference. In the final analysis what matters is that the overall process is clear. In this regard, inserted comments help legibility and subsequent processing.

Overall, the number of actions is reasonable although writing scripts with the Automation Suite certainly takes prac-

tice. Even though no special knowledge of a programming language is required, basic programming knowledge is still useful, especially with regard to a clear structure. The comprehensive, context-sensitive help is definitely very useful and contains examples as well as the description of the actions themselves. Moreover, installation of Management Suite provides some scripts that can be opened and viewed in Automation Suite, and the manufacturer has also published some videos on the topic, which provide a step-by-step guide.

### Easy Recording of Actions

If the procedures described above do not help to program a process, the final option is GUI automation via the interface action. The main procedure consists in recording a process like a program installation. For this purpose, Automation Suite is required on an end device that is used to create software installation packages. First, the application to be installed is started with

the action "External Process" via Automation Suite. Debug mode makes it easy to move from action to action when adding to the script. As soon as the first input window of the application appears, the administrator clicks anywhere in the window of Automation Suite to change the focus and then presses the F2 key. This hides Automation Suite and the Analyzer window appears at the top right of the screen.

If the administrator now places the cursor over any object on the screen, a red frame appears around the object. Detailed information about it, such as the window name and the position, is displayed in the Analyzer window. To record the rest of the installation, the administrator now moves to the next element to be operated. Instead of left-clicking, it is now important to right-click so that a window for recording the interface appears.

Detailed information can now be specified here if, for example, the standard installation path needs to be changed. If the information is correct, the administrator presses a button with a green arrow and a tick in the recording window, in order to generate and immediately play back the action. The next prompt then appears and so on, until the installation is complete. The administrator must now close the Analyzer window, which then causes the GUI of Automation Suite to open again. In the script window he sees then all of the recorded steps.

Next, he can add comments and any notes for dealing with errors, if necessary, so that later on it is clear whether or not the installation was successful. The saved script can now be reused within the Management Suite in order, finally, to generate installation requests. During the test we created shorter installation scripts and here the videos in particular were a great help. Once we had got the hang of things, we were able to create scripts quickly in this way. We liked the fact that the Analyzer actually records the individual objects and uses them for addressing, rather than cursor positions or similar. This makes the scripts robust so that they then also run on different end devices.

However, the administrator needs to realize that this method is a last resort, if other methods such as Application Wizard or the use of actions in Automation Suite won't work. If a setup is suitable for an automatic installation, Application Wizard requires far fewer steps than the method using GUI automation and is also much easier to set up.

Even if scripts generated with Automation Suite are mainly intended for further use in the Management Suite, they can also be used independently. To do so, Automation Suite couples the scripts with a runtime, thereby generating an executable EXE file which does not require a management agent to be started. According to baramundi, some customers are already using this.

### baramundi Automate: Conclusion

The Automate module is a valuable addition to baramundi Management Suite and designed to work with the other modules. However the scripts generated with Automation Suite can still be used independently, via a runtime addition. Functionally, the scope of commands is tailored specifically to software installations. However, the incorporation of external scripts and CMD commands makes it possible in principle to program very different processes. The administrator does not require programming knowledge, unless they wish to use these external scripts. Until then, programming is simply a matter of dragging & dropping and specification of parameters.

IT

#### baramundi Automate

##### Product

Program for the creation of software installation packages

##### Vendor

baramundi

[www.baramundi.com](http://www.baramundi.com)

##### Price

baramundi Automate costs €1500 per administration workplace, plus maintenance.

##### System requirements

baramundi Management Server for approx. 500 target systems to be managed:

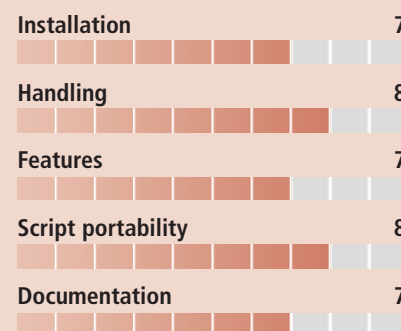
Current processor (dual or quad core, depending on the number of computers), minimum 2 GB RAM (4 GB recommended).

At least 1 GB hard disk space and space for the operating systems, patches, and applications to be deployed.

Operating system: Windows Server 2008 R2 SP1, 2008 SP2, 2012, and 2012 R2, PowerShell and .NET Framework 4.51.

baramundi database: MS SQL Server 2012 SP1, 2008 R2 SP2, or 2008 SP3 (in mixed mode), also MS SQL Express Version up to approx. 300 managed devices, Oracle as of Version 11 gR2 with latest service pack.

#### baramundi Automate IT-Administrator judges



Details of our testing methodology at  
[www.it-administrator.de/testmethodik](http://www.it-administrator.de/testmethodik)