

IT Administrator

Das Magazin für professionelle System- und Netzwerkadministration

Vergleichstest Skript-basierte
Softwarepaketierer:

Paketdienste im Wettbewerb

Vergleichstest Skript-basierte Softwarepaketierer

Paketdienste im Wettbewerb

von Jürgen Heyer



Tools zur Erstellung von Softwarepaketen demonstrieren eindrucksvoll, wie sich wiederkehrende Abläufe mittels Scripting automatisieren lassen. Die Herausforderung besteht allerdings darin, dass sich die Skripte möglichst automatisch erstellen und zudem einfach anpassen lassen. IT-Administrator hat sich dazu ACMP Automate, baramundi Automate und WinCM Scripting Framework genauer angesehen, die skript-basierte Softwarepakete erstellen. Wir wollten wissen, wie viele Programmierkenntnisse erforderlich sind und wie es um die Einsatzbandbreite bestellt ist.

Softwarepaketierung und Client-Management sind erfahrungsgemäß eng verwandt, sodass wir auf der Suche nach Scripting-Werkzeugen bei den Herstellern von Client-Management-Software schnell fündig wurden. Teilweise handelt es sich um Module, die zwar auch einzeln angeboten werden, letztendlich aber Bestandteil größerer Suiten zur Softwareverteilung sind. Getestet haben wir ACMP Automate von Aagon, das auch Bestandteil der ACMP Suite ist. Das zweite Produkt ist baramundi Automate, das in baramundi Deploy enthalten ist. Das gehört wiederum zur modular aufgebauten baramundi Management Suite. Der letzte Testkandidat ist das auf PowerShell basierende Scripting Framework der Windows Client Management AG (WinCM), die sich auf Softwareverteilung, Paketierung und Client/Server Engineering im Windows-Umfeld spezialisiert hat.

Wir wollen bei allen drei Werkzeugen die Scripting-Funktionalitäten betrachten in Hinblick darauf, wie leicht sich Skripte erstellen lassen, welche Programmierkenntnisse erforderlich sind, wie einfach Anpassungen erfolgen können und ob die Möglichkeiten über die Erstellung von Softwareinstallationspaketen hinausgehen.

WinCM Scripting Framework 1.1

Das "Scripting Framework" stammt von der Schweizer Firma "Windows Client Management" (WinCM) und ist aus einem größeren Projektauftrag entstanden. Um es gleich vorwegzunehmen, handelt es sich hier um diejenige der drei Lösungen, die am meisten Programmierkenntnisse verlangt. Scripting Framework ist ein auf PowerShell basierendes Toolkit, und wie die Bezeichnung Toolkit schon vermuten lässt, ist etwas Handarbeit erforderlich. Dies beginnt bereits bei der Einrichtung, die erfreulicherweise gut beschrieben ist und mehrere Möglichkeiten eröffnet.

Das Ziel von Scripting Framework ist das Erstellen und Ausführen von Installationspaketen auf normalen Clients, in Citrix-Umgebungen sowie in Verbindung mit Microsoft App-V. Dazu ist das Framework auf allen Clients und Terminalservern zu installieren, um die Pakete ausführen zu können. Die Installationspakete selbst werden auf einer zentralen Freigabe bereitgestellt, um sie so von überall im Zugriff zu haben. Für den Paketersteller gibt es zusätzlich die Möglichkeit, ein komplettes lokales Packaging-Umfeld zum Paketieren und Testen einzurichten, das so genannte Scripting Framework Workplace.

Verteilung des Toolkit auf mehreren Wegen

Prinzipiell kann das Scripting Framework eine Softwareverteilung weitgehend ersetzen, es kann aber auch mit einer bereits vorhandenen zusammenarbeiten. Insofern beschreibt das Handbuch die Installation des Frameworks manuell, per Soft-

wareverteilung, über das Microsoft System Center sowie über Gruppenrichtlinien. In unserer Testumgebung wählten wir den manuellen Weg.

Anfangs stutzten wir im Hinblick auf die ungewöhnlichen Setuproutinen, erkannten aber später, dass es sich hier bereits um CMD-Pakete handelt, die mit dem Toolkit erzeugt wurden. Sie bestehen jeweils aus einer Verzeichnisstruktur mit diversen Dateien. Die Voraussetzungen für den Einsatz des Toolkits sind das .NET Framework 4.0 sowie die PowerShell 3.0, wozu zwei fertige Pakete bei WinCM heruntergeladen werden können. Dann ist eine Freigabe für die Speicherung der Pakete einzurichten und ein Account auf diese Freigabe mit Lese-rechten im Active Directory anzulegen.

Vor der Installation des eigentlichen Scripting-Framework-Pakets ist innerhalb dessen ein Programm zur sogenannten Erstkonfiguration aufzurufen. Einzugeben sind der Freigabepfad und die Credentials des angelegten Accounts, außerdem können noch einige weitere Einstellungen angepasst werden. Ist dies erledigt, kann das Framework installiert werden. Auf den Clients, auf denen der Administrator Pakete entwickeln will, muss er nun noch ein weiteres Paket

installieren, die sogenannte Engineer-Erweiterung. Dann lassen sich Pakete direkt aus der Powershell ISE Console starten.

Im Anschluss an die Installation beschreibt das Handbuch ausführlich den Inhalt der Verzeichnisse des Toolkits sowie die angelegten Registry-Einträge. Anhand dessen lässt sich erkennen, ob alles korrekt eingerichtet wurde. Auch konnten wir über die Powershell ISE Console sehen, dass der zusätzliche Befehlsatz vorhanden war. Für die Konfiguration des Scripting Frameworks gibt es in der Registry einige Werte, die sich anpassen lassen. So lässt sich beispielsweise ein Reboot nach Paketausführung verhindern oder der Umfang der Benachrichtigungen festlegen.

Auf einem Client richteten wir nun das erwähnte Scripting Framework Work-place ein, um hier Pakete zu bauen. Neben dem Grundgerüst des Frameworks war dann auch hier eine Verknüpfung zur Powershell ISE Console vorhanden, darüber hinaus ein Packager für spezielle Aufgaben. Setzt der Administrator in der ISE Console den Befehl *Get-Command f_** ab, so erhält er eine lange Liste mit über 50 Befehlen, die durch das Scripting Framework zum PowerShell-Kommandosatz hinzukommen. Insgesamt erfordert die Ersteinrichtung diverse manuelle Schritte, und eine Verteilung auf die Clients sollte auf jeden Fall entsprechend der im Handbuch genannten Möglichkeiten automatisiert werden.

Einfache Veredelung von Standardpaketen

Die Intention von Scripting Framework besteht nicht darin, die üblichen Installationspakete von Applikationen zu ersetzen, sondern ein Konstrukt drumherum zu bauen, das durch eine Individualisierung noch mehr Komfort bietet und zugleich den vom Hersteller einer Applikation vorgesehenen Setup-Weg unangetastet lässt. Das bedeutet, dass das Framework ein Repackaging mit all seinen Problemen umgeht, denn bei einem Repackaging geht in der Regel der Support durch den Hersteller verloren. Scripting Framework geht vielmehr von der regulären Silent-Installation einer Applikation aus. Der Administrator muss also zuerst ermitteln, welche Befehlssequenz für eine Silent-Installation erforderlich ist. Das ist jedoch

meist inklusive eventueller Optionen gut dokumentiert.

Ein Paket bei Scripting Framework besteht aus einem Verzeichnis mit einer bestimmten Unterstruktur und vorgegebenen Inhalten. Mit der Installation kommt ein Template, von dem der Administrator eine Kopie erzeugt, die er dann mit den individuellen Inhalten füllen kann. Zuerst benennt er das Template nach einem vorgegebenen Muster um und füllt die Datei "Package.xml" teils mit ähnlichen Werten, um das Paket genau und einheitlich zu beschreiben. Inhalt sind unter anderem der Hersteller der Software, die Build-Nummer des Pakets, die eigene Firma oder ein Standort, Sprache und Version. Das Verzeichnis für ein Installationspaket von Adobe Photoshop Elements könnte beispielsweise den Namen "Adobe_Photoshop_Elements_11_UNV_MUI_01" haben.

UNV bedeutet hier, dass es sich um ein globales Paket handelt, und MUI steht dafür, dass die Sprache erst bei der Paketausführung analog zum Client festgelegt wird. Indem die Namensgebung klar definiert ist, lassen sich auch viele Pakete noch erfreulich übersichtlich verwalten, auch von mehreren Administratoren. In einem Paketverzeichnis befinden sich nun auf der ersten Ebene die Dateien "Install.exe" sowie "Uninstall.exe" mit den beiden zugeordneten PowerShell-Skripten "Install.ps1" sowie "Uninstall.ps1". Diese werden entsprechend von den beiden EXE-Dateien aufgerufen und enthalten die eigentlichen Skriptanweisungen. Parallel dazu gibt es die Ordner "Classic" und "AppV" zur Unterteilung in eine klassische Programminstallation sowie eine virtuelle Anwendung. Jeweils darunter existieren drei Unterverzeichnisse "Config" mit der Konfigurationsdatei zum Paket, "Setup" mit den Quelldateien der zu installierenden Applikation und "User" mit benutzerspezifischen Dateien.

Die beiden PS1-Dateien sind mit diversen häufig verwendeten Framework-Befehlen gefüllt, um beispielsweise für eine Installation automatisch die Sprache des Clients zu ermitteln und Ladevariablen auszuwerten, aber auch, um die Installation der eigentlichen Applikation anzustoßen. Der Administrator muss nun diesen Inhalt

WinCM Scripting Framework 1.1

Produkt

Auf der PowerShell basierende Befehls-satzerweiterung zum Erstellen von Softwarepaketen sowie zum Ausführen von sonstigen Skripten.

Hersteller

Windows Client Management AG
www.wincm.ch

Preis

Die Lizenzierung erfolgt auf jährlicher Subskriptionsbasis entsprechend der Anzahl der Clients, 100 Clients kosten rund 2300 Euro, 200 Clients 2575 Euro, für noch größere Mengen gibt es weitere Staffelpreise.

Systemvoraussetzungen

Microsoft .NET Framework 4.0 oder höher, PowerShell 3.0 oder höher, eine zentrale Freigabe für die Konfigurationsdateien mit einem leseberechtigten Service Account.

Technische Daten

www.it-administrator.de/downloads/datenblaetter

abändern und nicht benötigte Zeilen ausdokumentieren oder löschen. So entsteht mit relativ wenig Aufwand das fertige Installationsskript. Selbstverständlich kann der Administrator auch auf sämtliche PowerShell-Befehle zurückgreifen, sofern er komplexe Abläufe programmieren will. Auch lässt sich das Scripting Framework mit entsprechenden PowerShell-Programmierkenntnissen für ganz andere Automatisierungen verwenden.

Mehr Effizienz durch geschickte Variablen-Nutzung

Innerhalb von Scripting Framework steht eine Vielzahl an Variablen zur Verfügung (Maschinenvariablen, Maschinen- sowie Benutzer-Laufzeitvariablen, Benutzervariablen lokal sowie aus dem Active Directory), die es erlauben, dynamische Pakete zu erstellen. Auch gibt es neben einer globalen Konfigurationsdatei optional eine standortspezifische. Beim Laden von Variablen wird automatisch überprüft, ob eine standortspezifische CFG-Datei vorhanden ist, aus der der Standort ausgelesen wird. Dementsprechend werden die globalen oder standortspezifischen Variablen verwendet. Eine Grafik im Handbuch zeigt genau auf, welche Logik implementiert ist.

Durch die Verwendung dieser Variablen in den Skripten lassen sich diese nun dynamisch anpassen, sodass es möglich ist, für mehrere Standorte ein Skript zu verwenden, das dann doch jeweils individuell abläuft mit unterschiedlichen Einstellungen. Das resultiert letztendlich vor allem in größeren und verteilten Umgebungen in weniger Paketen, was gerade bei Updates von Applikationen, die eine Neuerstellung der Pakete erfordern, eine enorme Einsparung bedeutet. Auch ist es möglich, zwischen 32- und 64-Bit-Umgebungen zu unterscheiden, sodass hier nicht doppelte Pakete erstellt werden müssen. Das einheitliche Format erleichtert zusätzlich die Pflege.

Eine weitere Einsparung bei der Anzahl der notwendigen Pakete resultiert aus der Ausführung eines benutzerspezifischen Skriptteils, der ebenfalls Einstellungen dynamisch anpasst. Dieser Teil wird dann ausgeführt, wenn sich ein Benutzer anmeldet. Angesichts der möglichen Dynamik beim Paketablauf spricht der Her-

steller von Hybrid-Paketen und es ist möglich, das gleiche Paket auf Servern inklusive Citrix sowie Clients gleichermaßen einzusetzen.

Statt die über das Scripting Framework verfügbaren Befehle nur in fertigen Skripten zu nutzen, kennt die Powershell ISE Console auch einen so genannten Scripting-Modus. Hierbei kann der Administrator in der Konsole Befehle eingeben und sofort ausführen lassen. Das erlaubt eine schrittweise Abarbeitung mit sofortiger Beobachtung des Resultats.

Paketbau aus den Unterschieden

Wenn es darum geht, beispielsweise benutzerspezifische Einstellungen für ein Ausrollen über das Scripting Framework zu erfassen, dann stellt der Scripting Framework Packager, der mit der Workplace-Variante kommt, eine gute Möglichkeit zur Aufzeichnung dar. Der Packager arbeitet im Delta-Verfahren, indem er die Unterschiede per Snapshot-Verfahren aufzeichnet. So kann der Administrator eine Applikation zuerst mit den globalen Einstellungen verteilen. Dann wird der Packager aufgerufen und ein erster Snapshot erstellt. Dabei lässt sich auswählen, welche Teile der Registry aufgezeichnet werden und ob der Inhalt von Laufwerk "C:" aufgenommen werden soll.


Anschließend ruft der Administrator die Applikation auf und führt die benutzerspezifischen Änderungen durch, dann erstellt er einen zweiten Snapshot. In nächsten Schritt vergleicht der Packager die Einstellungen und erzeugt aus den ermittelten Unterschieden ein komplettes Installationspaket. Dieses lässt sich nun weiter anpassen und über das Scripting Framework nutzen.

Um bei Problemen bei der Skripterstellung den Grund leichter zu finden, hat der Hersteller in das Framework ein umfassendes Logging integriert. Dieses unterscheidet zwischen einem Computer-Log, einem Benutzer-Log und Kopieraktionen. Beim Erreichen von einem MByte Größe wird das jeweilige Log archiviert und ein neues begonnen, damit die Länge nicht die Performance beeinflusst. Beim Betrachten der Logs wird erkennbar, dass die Aufzeich-

nung sehr detailliert erfolgt. Die Paketausführung im Framework beendet sich mit Return Codes, die MSI-konform sind, was eine leichtere Weiterverarbeitung erlaubt. Die Return Codes finden sich auch im Scripting Framework Log wieder.

Fazit Scripting Framework

Das PowerShell-basierte WinCM Scripting Framework erweitert den Befehlssatz um über 50 Anweisungen zur Erstellung von Softwarepaketen. Ziel ist es dabei, kein Repackaging von Paketen der Applikationshersteller durchzuführen, sondern die Funktionalität zu erweitern. Ein deutlicher Mehrwert ergibt sich durch die dynamische Anpassung von Einstellungen unter anderem aufgrund der Lokation und der Betriebssystemsprache des Clients. Auch können Pakete gleichermaßen für Citrix-Umgebungen, Server und Arbeitsplätze verwendet werden. Weiterhin lassen sich benutzer-spezifische Anpassungen realisieren.

Die Einrichtung des Scripting Frameworks erfordert allerdings ein wenig Handarbeit. Dank diverser Vorlagen und Beispiele lässt sich der Umgang mit dem Framework recht schnell erlernen. Programmierkenntnisse in der PowerShell sind nicht unbedingt erforderlich, aber hilfreich, um den vollen Leistungsumfang nutzen zu können. Wer die PowerShell beherrscht, kann auch beliebige eigene Skripte erstellen und über das Framework auf den Zielsystemen ausführen lassen. Für eine schnellere Inbetriebnahme und Einarbeitung bietet der Hersteller sowohl Projektunterstützung als auch Schulungen an. (jp) 

Scripting Framework
So urteilt IT-Administrator

Installation	5
Bedienung	5
Funktionsumfang	9
Skript-Portabilität 9	
Dokumentation	7

Die Details unserer Testmethodik finden Sie unter www.it-administrator.de/testmethodik