

# **b4A** White Paper

Package Management – Der Weg zu einem Deployment-Verfahren

**best-blu consulting with energy GmbH**

Ihr Ansprechpartner:

**Andreas Büsching**

E-Mail: [a.buesching@best-blu.de](mailto:a.buesching@best-blu.de)

Telefon: +49 (0) 5341 85393-0

Mobil: +49 (0) 174 32985-14

Datum: 24. November 2020

## INHALT

Motivation.....	3
Ist-Situation .....	3
Namenskonventionen .....	3
Entwicklungsrichtlinien (Compliance) .....	4
Konfiguration .....	4
Deployment.....	4
Reproduzierbarkeit.....	5
Zusammenfassung .....	5
Lösung .....	5
Migration .....	6
Von Anfang an .....	6
Side by Side.....	6
Neue Umgebung.....	6
Neuer Mandant.....	6
Step by Step .....	6
Fazit .....	7

## Motivation

Der Bedarf an Automation ist in den letzten Jahren enorm gewachsen. Dabei hat sich nicht nur die Menge der automatisierten Prozesse erhöht, sondern auch die Anforderungen sind gestiegen und damit ebenso die Komplexität.

Viele der Automationsumgebungen auf Basis von Atomic Automation sind seit vielen Jahren in den Unternehmen und versuchen sich den wachsenden Bedürfnissen und Anforderungen zu stellen. Mit der heutigen Komplexität kommen viele der Umgebungen an ihre Grenzen. Gründe hierfür sind vielseitig. Ein Punkt, der in der Regel dazu gehört, ist die wachsende Wichtigkeit und damit Bedeutung der Automationssysteme und der darauf laufenden Prozesse. Fehler in den Prozessen oder andere Instabilitäten dieser Umgebungen haben mittlerweile oft fatale Auswirkungen auf die unterschiedlichsten Bereiche eines Unternehmens.

Um aus alten gewachsenen Automationsinfrastrukturen und den zugehörigen Entwicklungs- und Testverfahren eine Umgebung zu bauen, die diesen neuen Anforderungen gerecht wird, bedarf es größeren Änderungen. Diese müssen in verschiedenen Gebieten angewendet werden, um eine moderne, stabile und damit qualitativ hochwertige Automationsplattform zu schaffen.

Dieses White Paper beschäftigt sich mit einem dieser Themengebiete und zeigt hierfür eine Lösung auf. Dabei geht es um ein Konzept, das eine Grundlage für eine strukturierte Entwicklung mit wohldefinierten Prozessen festlegt und zusätzlich ein Verfahren vorstellt, mit dem ein vollautomatisiertes Deployment-Verfahren möglich ist.

## Ist-Situation

Viele Umgebungen befinden sich schon viele Jahre in der Infrastruktur und versuchen sich den ständigen Änderungen und wachsenden Anforderungen anzupassen. Hierbei sind die Weiterentwicklung von Standards in der Automation häufig auf der Strecke geblieben, wodurch ein Mithalten immer schwieriger wird.

Um qualitativ gute und immer komplexere Automationsprozesse möglichst schnell in die Produktion zu bringen, bedarf es neuer Konzepte in der Entwicklung, beim Testen und für ein Deployment-Verfahren. Diese müssen sich den Herausforderungen und Problemen widmen, die heutzutage in vielen Atomic Automation Umgebungen existieren. Im Folgenden werden einige Beispiele aufgezeigt, um die Intentionen dieser Konzepte zu verdeutlichen.

### **Namenskonventionen**

Regeln für die Namen der Objekte sind in einer Atomic Automation Umgebung in erster Linie wichtig, um ein Berechtigungssystem aufzusetzen. Des Weiteren sind Objektnamen eindeutig pro Mandanten und sollten Informationen über die Funktion und logische Zugehörigkeit eines Objektes eine Aussage treffen. Aus diesen Gründen sind Fehler in den Objektnamen gegebenenfalls nicht nur ein

Schönheitsfehler, sondern können kritischere Auswirkungen haben. Daher ist es wichtig, dass Objektnamen korrekt sind und niemals fehlerhaft benannte Objekte bis in die Produktion gelangen.

Tippfehler bei der Erstellung oder Umbenennung von Objekten sind eine häufige Ursache für solche Fehler. Da in der Regel auch kein Review auf der Ebene stattfindet, werden solche Fehler selten entdeckt.

### **Entwicklungsrichtlinien (Compliance)**

Um Anforderungen in einer Automatic Automation Umgebung umzusetzen, gibt es in der Regel viele verschiedene Möglichkeiten. In vielen Umgebungen findet man daher auch die unterschiedlichsten Lösungsvarianten für ein und dasselbe Problem. Für Entwickler heißt dies, dass sie sich öfter als notwendig in Lösungen einarbeiten müssen, die von anderen entwickelt worden sind. Solch eine „freie“ Form der Entwicklung führt allerdings nicht nur in der Entwicklung zu erhöhten Aufwänden, sondern kann auch das Lösen von kritischen Produktionsproblemen erschweren.

### **Konfiguration**

In den Umgebungen, wo bereits Produktion und Entwicklung getrennt sind, d.h., dass es mehr als eine Automatic Automation Umgebung gibt, wird die Schwierigkeit bezüglich der unterschiedlichen Konfigurationen bekannt sein. Klassische Fälle, die hier häufig auftreten, sind unterschiedliche Pfade auf dem Dateisystem, andere Namen für die Agenten und andere E-Mail-Adressen. Es gibt allerdings noch deutlich mehr Varianten. Beispiele Hostnamen oder URLs zu externen Systemen, die mit der Automatic Automation angesteuert werden. Vermutlich die meisten Lösungen für diese Herausforderung basieren darauf, in irgendeiner Form die exportierten Objekte zu ändern. Dafür kommen eigens geschriebene Skripte zum Einsatz oder die Änderungen werden jedes Mal manuell nach dem Import durchgeführt. Eine andere Variante ist die Verwendung von komplexeren Include-Objekten, die zur Laufzeit basierend auf der Umgebung, in der sie laufen, die Konfiguration anpassen.

All diese Lösungen weisen in der Praxis Risiken auf, die durch falsche Änderungen die Stabilität der Produktion riskieren oder zur Laufzeit zusätzliche Ressourcen benötigen und die Entwicklung erschweren.

### **Deployment**

Alle Punkte, die zuvor genannt worden sind, können ein stabiles Deployment der Objekte beeinflussen. Es gibt allerdings noch weitere Punkte, die ein Deployment gegebenenfalls zu einer gefährlichen Operation werden lassen können. Entscheidend bei einem Deployment von Objekten aus der Automatic Automation ist die Auswahl der richtigen Objekte. Werden einige der Objekte auch noch von anderen Prozessen genutzt, ist sicherzustellen, dass diese die Kompatibilität zu anderen Objekten in der Produktion nicht brechen. Da heutige Automationsumgebungen viele zehntausend oder gar hunderttausende von Objekten haben, ist es kaum noch möglich, hier den Überblick zu bewahren. Dass für ein Deployment zu wenige, zu viele oder gar die falschen Objekte vorgesehen werden, tritt durchaus auf. Die Häufigkeit hängt von mehreren Faktoren ab. Dazu gehören die Komplexität der Automationsprozesse sowie die Prozesse in der Entwicklung an sich.

### Reproduzierbarkeit

Dieses Stichwort dürfte heutzutage in allen IT-Abteilungen fallen und ein Grundsatz bei jedem Update sein. Dabei geht es zusätzlich in vielen Fällen auch noch darum, den Urzustand in möglichst kurzer Zeit wiederherzustellen. Mit den Werkzeugen der Atomic Automation und dem integrierten Version Management ist eine Wiederherstellung der alten Versionen durchaus möglich. Dies hat allerdings seine Grenzen. Jedes der Objekte hat seine eigene Versionsnummer und manche Objektarten werden häufig durch manuelle Nacharbeiten verändert (bspw. Variablen und Schedules). Somit ist eine Wiederherstellung machbar, kann aber zu einer komplexeren Aufgabe werden.

### Zusammenfassung

Zwischen den zuvor beschriebenen Problemen gibt es eine klare Gemeinsamkeit. Alle diese Probleme stehen im Zusammenhang mit manuellen Arbeiten, die keiner oder einer unzureichenden Kontrolle unterzogen werden.

## Lösung

Das best4Atomic Package Management definiert ein Konzept, dass die Objekte in logische Gruppen einordnet und diese Gruppen versioniert. Solche Gruppen werden *b4A Packages* genannt. Auf dieser Basis werden Funktionalitäten angeboten, die alle der oben beschriebenen Probleme adressiert und dabei auch noch weitere Fähigkeiten beherrscht.

Beispielsweise stellt b4A ein Modul bereit, mit dem sowohl die Namenkonvention als auch die Entwicklungsrichtlinien überprüft werden können. Durch den integrierten Konfigurationsmechanismus können Entwickler für alle Umgebungen und, wenn gewünscht, auch für einzelne Mandanten Konfigurationen definieren. Die jeweils korrekte wird beim Deployment automatisch in den Mandanten eingespielt. Für das Deployment müssen jetzt nur noch *b4A Packages* ausgewählt werden und keine einzelnen Objekte. Durch Abhängigkeiten zwischen den *b4A Packages* wird sichergestellt, dass die Installation in einem Zielmandant keinen Schaden anrichtet und auch alle notwendigen Objekte vorhanden sind.

Mit dem best4Atomic Package Management ist es jeder Zeit möglich, auf eine alte Version zurückzugreifen und somit den exakten Zustand wiederherzustellen, der vor einem Update existierte.

Durch den Konfigurationsmechanismus und zusätzliche Fähigkeiten werden einige typische Post-Deployment-Aufgaben abgedeckt. Es können beispielsweise Schedule-Objekte aktualisiert und das Neuladen der Definition angestoßen werden. Veralterte Objekte werden automatisch identifiziert, umbenannt und verschoben. Zugangsdaten von Login- und Connection-Objekten können bei der Installation wiederhergestellt werden und können somit ebenfalls transportiert werden. Und zusammen mit weiteren Funktionen bietet dieses Verfahren Lösungen für viele Herausforderungen im Bereich des Deployment von Automationsprozessen.

## Migration

Da jede Umgebung anders ist und die Ausprägungen der Automatisierung sich ebenfalls unterscheiden, gibt es nicht nur einen Weg zum Package Management. Im Folgenden werden grundlegende Strategien aufgezeigt, die bei einer Migration verfolgt werden können. Alle diese Formen wurden bereits in Kundenprojekten umgesetzt.

### **Von Anfang an**

Die einfachste Migration ist keine Migration. Wenn die Automationsumgebung gerade erst aufgesetzt wird, dann ist das Wichtigste, die Grundlagen und Konfigurationen für das Package Management ganz klar und detailliert zu definieren. Hierzu gehören auch Entwicklungsrichtlinien und Prozesse, die mit der Anforderung beginnen und erst mit dem Deployment in der Produktion enden. Klare Richtlinien mit möglichst wenigen Ausnahmen sind der wichtigste Faktor für eine optimale und effiziente Durchführung.

### **Side by Side**

Dies ist eine der am häufigsten genutzten Strategien. Automationsumgebungen, die mit den Jahren viele Probleme gesammelt haben, brauchen, wenn möglich, einen Neuanfang. Dieser stellt sicher, dass Altlasten nicht weiter bestehen und somit einem Neuaufbau nicht im Weg stehen. Diese Strategie gibt es in zwei Ausprägungen, und beide Verfahren benötigen in der Regel kaum zusätzliche Ressourcen.

### **Neue Umgebung**

Häufiger in Kombination mit einem Upgrade-Projekt wird die Chance genutzt, nicht nur die Automation Engine neu aufzusetzen, sondern auch die Entwicklung auf komplett neuen Standards aufzubauen. Dabei werden ab einem Zeitpunkt sämtliche Neuentwicklungen nur noch auf Basis des b4A Package Management durchgeführt und nur die Maintenance alter Prozesse wird noch in der alten Umgebung mit alten Verfahren durchgeführt. Schrittweise werden dann die alten Prozesse in die neue Umgebung gezogen und dabei in b4A Packages verwandelt.

### **Neuer Mandant**

Diese Technik ähnelt der obigen Variante sehr stark. Nur in diesem Fall wird nicht eine komplett neue Umgebung genutzt, sondern nur ein weiterer Entwicklungsmandant eingerichtet, der nach dem b4A Package Management Konzept arbeitet. Die Art und Weise der Migration bleibt gleich. Dieses Verfahren erleichtert die Synchronisation zwischen Prozessen in der alten und der neuen Welt.

### **Step by Step**

Bei dieser Strategie werden nur langsam die neuen Konzepte und Verfahren in demselben Mandanten eingeführt, in dem auch die alten Techniken zum Einsatz kommen. Allerdings werden die alten Objekte auf einmal in die neuen Ordnerstrukturen überführt. Dafür werden die Objekte in einem Klon-Mandant in die neuen Package-Ordnerstrukturen überführt. Hier muss eine umgebungsspezifische Lösung gefunden werden, die mit Unterstützung von b4A umgesetzt werden

kann. Anschließend kann mit Hilfe eines b4A Moduls die neue Struktur in dem Hauptentwicklungsmandant nachgebildet werden. Nach diesen Schritten sind die Objekte zwar in die neuen Ordnerstrukturen überführt worden, aber die Namenkonventionen stimmen noch nicht, und auch einige weitere Fähigkeiten des Package Management können so noch nicht genutzt werden. Die restlichen Anpassungen erfolgen schrittweise und können beispielsweise immer dann angegangen werden, wenn die Objekte eines Packages durch Anforderungen geändert werden müssen.

## Fazit

Das best4Automatic Package Management stellt eine stabile, wohldefinierte Basis für moderne Automatic Automation Umgebungen dar, die heutige häufig auftretende Probleme in Automationsumgebungen angeht und Lösungen dafür zur Verfügung stellt. Diese sind stark konfigurierbar und bieten so die notwendige Flexibilität, um sich an jede Umgebung anzupassen.