

Packaging Guide

Informationshandbuch

Produktversion 7.6

columbus

Stand: 12.18

© brainwaregroup - 1997-2017 - Alle Rechte vorbehalten

Jedwede von der brainwaregroup zur Verfügung gestellte Dokumentation unterliegt dem Urheberrecht und ist Eigentum der brainwaregroup. Die brainwaregroup übernimmt weder Garantie noch die juristische Verantwortung oder irgendeine Haftung für die Nutzung dieser Informationen, für deren Wirtschaftlichkeit oder fehlerfreie Funktion für einen bestimmten Zweck.

Bei Zusammenstellung dieses Dokuments wurde jede Anstrengung unternommen, die Richtigkeit des Inhalts sicherzustellen. brainwaregroup übernimmt jedoch keine Garantie hinsichtlich dieser Dokumentation und keine gesetzliche Gewährleistung für die marktgängige Qualität und Eignung für einen bestimmten Zweck. Des Weiteren übernimmt brainwaregroup keine Haftung für Fehler oder unbeabsichtigte Schäden bzw. Folgeschäden im Zusammenhang mit der Bereitstellung, Leistung oder Verwendung dieses Dokumentes oder der darin enthaltenen Beispiele. brainwaregroup behält sich das Recht vor, diese Dokumentation ohne vorherige Ankündigung zu ändern.

Alle in diesem Dokument in Beispielen verwendeten Namen, Firmennamen oder Firmen sind fiktiv und beziehen sich weder im Namen noch im Inhalt auf tatsächliche vorhandene Namen, Organisationen, juristische Personen oder Institutionen oder sollen diese darstellen. Jegliche Ähnlichkeit mit vorhandenen Personen, Organisationen, juristische Personen oder Institutionen ist rein zufällig.

Die in diesem Dokument beschriebene Software wird unter den Bedingungen eines Lizenzvertrags zur Verfügung gestellt und darf nur im Einklang mit den Bestimmungen dieser Vereinbarung verwendet werden.

Dokumententitel	Packaging Guide - Informationshandbuch
Produktversion	7.6
Herstellung und Druck	Brainware Consulting & Development AG Sumpfstrasse 15 CH-6300 Zug
Veröffentlichungsdatum	12.12.2018

Dieses Dokument darf ohne vorherige schriftliche Genehmigung der brainwaregroup, weder vollständig noch auszugsweise kopiert, fotokopiert, reproduziert oder weiterverarbeitet werden.

Inhalt

0	Allgemeines	6
0.1	Dokumentenhistorie	6
0.2	Ergänzende Dokumente.....	6
0.3	Änderungen gegenüber Packaging Guide 1.3.x.....	7
1	Grundlagen zu SW-Paketen	8
1.1	Maschinen- und Benutzerteil.....	8
1.2	Installationsreihenfolge der SW-Pakete.....	8
1.3	Verarbeitung innerhalb des SW-Paketes	9
1.4	SW-Pakete im Einsatz für Shareless.....	9
1.5	Konventionen für SW-Pakete	10
1.5.1	Nummerierung (Identifizier).....	10
	Identifizier (zentral)	11
	Identifizier (Aussenstelle)	12
	Identifizier (zentral basierend)	13
1.5.2	Description (zentral).....	13
1.5.3	Description (Aussenstelle)	14
1.5.4	Sprachcodes.....	14
	Sprachcodes für SW-Pakete	14
	Sprachcodes im Benutzerteil	14
	Sprachcodes im Maschinenteil.....	14
1.5.5	Version.....	15
1.5.6	Patches	15
1.5.7	Plattform (Komplexität)	15
1.5.8	Status.....	16

2	SW-Pakete bearbeiten	17
2.1	Package Definition / Configuration	18
2.2	Delivery-Script.....	20
2.3	Package Documentation	21
2.4	Configuration Script	22
2.5	Browse / Check Files in Package	23
2.6	Remove in SW-Paketen.....	23
2.7	SW-Paket Columbus Packaging Tools	24
2.8	Zusätzliche Werkzeuge in PackageStudio	24
3	Details zur SW-Paketierung	25
3.1	SW-Paketierungstechnik.....	25
3.2	Arbeiten mit „Packaging Tools“	25
3.3	Columbus Scriptsprache	26
3.4	Speicherort der Installationsquelle	26
3.5	Verknüpfungen	26
3.6	Konfiguration von SW-Paketen	26
3.6.1	Namenskonvention von Variablen.....	27
3.6.2	Aufbau einer Konfigurationsdatei.....	27
3.6.3	Variable Werte in einem Paket.....	28
3.7	Variablenauflösung - Columbus Scriptsprache	28
3.8	Informationen zu SW-Paketen.....	28
3.9	Repetitive Sektionen.....	29
3.9.1	Ausführung der Sektionen.....	29
3.9.2	Herstellung der Sektionen	29
3.9.3	Anwendungsbeispiele	29
3.10	Platzhalterpakete.....	30
3.10.1	Namenskonventionen von Platzhalterpaketen.....	30
3.10.2	Vorgehen zur Erstellung eines Platzhalterpaketes.....	30
4	Variablenwerte über die Konsolen-Struktur	31
4.1	Information.....	31
4.2	Kurzübersicht.....	31
4.3	Detailerläuterung je Variable.....	32
4.3.1	c_ConfigPath.....	32
4.3.2	c_MachineType.....	32
4.3.3	c_MachineLocation	32
4.3.4	c_MachineLanguage.....	33
4.3.5	c_MachineDep	33
4.3.6	c_UserDep.....	33
4.3.7	c_UserLanguage	34
4.4	Variablen pro Ebene.....	34
4.4.1	Auf Company Ebene.....	34
4.4.2	Auf Location Ebene	35
4.4.3	Auf Abteilung (Department) Ebene.....	35
4.4.4	Auf Maschinentyp Ebene	35
4.4.5	Auf Usertyp Ebene	36

4.5	Variablenwerte vom Management Client	36
5	Log- und Konfigurationsdateien	37
5.1	Erstellung von Logdateien	37
5.2	Speicherort von Logdateien	37
5.3	Speicherort von Konfigurationsdateien	38
5.4	Spracheinstellung von SW-Paketen	38
6	Software Abhängigkeiten	39
7	Setuproutinen	40
7.1	Columbus eigener MSI Befehl	41
7.2	Mit Windows Installer	41
7.3	Windows Installer Fehlercodes	42
7.4	Weitere Setuproutinen	43
7.5	Reboot on demand	43
8	QA - Package Engineer	44
8.1	Allgemein	44
8.2	SW-Paket Eigentest - Vorgehen	45
8.3	SW-Paket Eigentest – Häufige Fehler	47
9	Info.rtf Vorlage	48
10	SW-Releasemanagement	49
10.1	Prozessgrafik (Beispiel)	50
10.2	Prozessrollen	51
11	Glossar	52
11.1	Kürzel	52
11.2	Begriffe	52

Allgemeines

In diesem Kapitel

Dokumentenhistorie.....	6
Ergänzende Dokumente	6
Änderungen gegenüber Packaging Guide 1.3.x	7

0.1 Dokumentenhistorie

Version	Datum	Autor	Beschreibung
2.0.2	18.06.12	B. Nyffenegger	Freigabe an QA
2.0.1	28.08.12	B. Nyffenegger	Detaillkorrektur
2.0	18.07.12	B. Nyffenegger	Komplette Überarbeitung
1.3.2	15.02.11	H. Bergens	+ Kapitel <i>Reboot on demand</i>
1.3.1	07.06.10	B. Nyffenegger	(Freigabe) + Textüberarbeitung + Kapitel <i>Dummy oder Platzhalterpakete</i> ergänzt
1.3	10.03.10	B. Nyffenegger	(Freigabe) + Kapitel <i>Ergänzende Dokumente</i> + Kapitel <i>Gültigkeitsbereich</i> + Änderung an Namenskonvention von Log- und Konfigurationsdateien. + Die Informationen der Info.rtf wurden erweitert. + Kapitel <i>Variablenauflösung</i> + Kapitel <i>Informationen zu SW-Paketen</i> + Kapitel <i>Zusätzliche Werkzeuge in PackageStudio</i>
1.2	21.01.10	B. Nyffenegger	(Freigabe) + Kapitel <i>Dummy</i> resp. Platzhalterpakete + Kapitel <i>Repetitive Sektionen</i> - %c_MachineCostCenter% Variable entfernt
1.1	15.09.09	B. Nyffenegger	(intern) + Designanpassungen, zusätzliche Grafiken, Kapitel <i>Repetitive Sektionen</i> - %c_MachineCostCenter% entfernt
1.0	24.03.09	B. Nyffenegger	Freigabe an Projekt (extern)
0.0	28.09.07	B. Nyffenegger	Ersterstellung (intern)

0.2 Ergänzende Dokumente

Für weiterführende und ergänzende Informationen zu Packaging Guide verweisen wir auf folgende Handbücher:

- Columbus **Technical Reference**
Beschreibung der Scriptbefehle und Variablen sowie eine Übersicht der Konfigurationsparameter der Management Console (nur in englischer Sprache verfügbar).

0.3 Änderungen gegenüber Packaging Guide 1.3.x

Die nun vorliegenden Paketrichtlinien wurden mit grösster Sorgfalt und Rücksicht auf bestehende Projekte erstellt. Durch die Weiterentwicklung der Umsysteme, Bedürfnisse und Möglichkeiten zum Einsatz von Columbus (z.B. Shareless) war eine Reformation der Paketrichtlinien unumgänglich.

Die grösste Änderung betrifft die Verwendung der Maschinenteile (Server und Client) eines SW-Pakets:

Neu werden alle Vorgänge innerhalb des Server Scriptteils verarbeitet. Damit ist gewährleistet, dass nach Beendigung des Server Scriptteils, die Applikation komplett fertig installiert und betriebsbereit ist.

Der Client Scriptteil wird nicht mehr verwendet. Benutzereinstellungen werden auf das Minimum reduziert und wie bisher im User Scriptteil verarbeitet.

Hinweis

Diese Änderung betrifft keine mit der Snapshot-Technik hergestellten SW-Pakete. Diese werden wie bisher behandelt (Delivery, Configuration).

SW-Pakete werden nun in drei Komplexitätsstufen unterteilt. Diese Stufen werden als „Plattform“ bezeichnet:

EASY: Solche SW-Pakete werden für den „standalone“ Einsatz erstellt und können in jeder Columbus Umgebung ohne jeglichen Aufwand verwendet werden. Es werden keine Konfigurationseinstellungen (Ausnahme ist das Deaktivieren von Autoupdate Mechanismen) vorgenommen und keine Verknüpfungen oder dergleichen verändert.

ADVANCED: Diese SW-Pakete bieten die Möglichkeit, Einstellungen über eine Konfigurationsdatei anzusteuern. Die Konfigurationsdatei kann entweder zentral (%C_ConfigPath%) oder dezentral abgelegt werden. Es werden keine weiteren Aufwände oder Voraussetzungen seitens Columbus Umgebung benötigt. Es werden keine zusätzlichen, zentralen Variablen verwendet.

EXPERT: In dieser Plattform können alle verfügbaren Scripttechniken, Variablen (Kundenseitig etc.) verwendet und genutzt werden. Solche SW-Pakete sind in der Regel sehr kundenabhängig und bedürfen eines erweiterten Wissens über die Paketieretechnik zum einen und der Kundenumgebung zum anderen.

Das Ziel ist eine deutliche Entschlankung und verminderte Komplexität dafür erhöhte Interoperabilität von SW-Paketen über mehrere SW-Releases.

KAPITEL 1

Grundlagen zu SW-Paketen

In diesem Kapitel

Maschinen- und Benutzerteil	8
Installationsreihenfolge der SW-Pakete	8
Verarbeitung innerhalb des SW-Paketes	9
SW-Pakete im Einsatz für Shareless	9
Konventionen für SW-Pakete	10

1.1 Maschinen- und Benutzerteil

SW-Pakete bestehen aus einem Maschinen- und Benutzerteil. Die beiden Teile werden mit unterschiedlichen Benutzer-Accounts ausgeführt:

- Maschinenteile einerseits mit einem System-Account des Columbus Service, andererseits mit einem lokalen Admin-Account.
- Benutzerteile werden im Kontext des jeweils angemeldeten Benutzers ausgeführt.

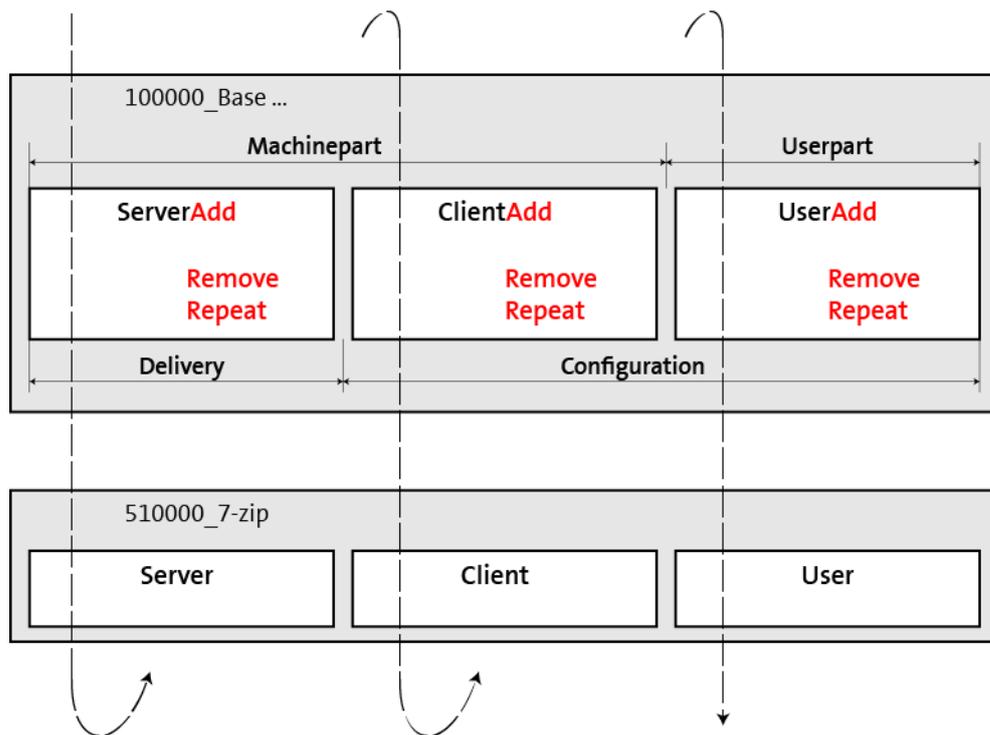
Der Maschinenteil ist unterteilt in einen Server- und Client-Teil.

1.2 Installationsreihenfolge der SW-Pakete

Nach Zuweisung eines SW-Paketes in der Management Console, wird der Management Client gepusht um die Neuzuweisung aus der Datenbank zu erhalten. Daraufhin beginnt er mit der Abarbeitung der SW-Pakete mit folgender Reihenfolge:

- Alle Server-Teile der zugewiesenen SW-Pakete.
- Alle Client-Teile der zugewiesenen SW-Pakete (wenn der dazugehörige Server-Teil erledigt ist).
- Alle User-Teile der zugewiesenen SW-Pakete (wenn der dazugehörige Client-Teil erledigt ist).

Innerhalb der Paket-Teile können gewollte oder ungewollte EXIT Markierungen auftreten. Diese Teile werden nach einem Neustart oder Clientpush wieder gestartet. Die Meldung über einen EXIT wird in die Management Console ebenso übermittelt wie eine erfolgreiche Beendigung.



1.3 Verarbeitung innerhalb des SW-Paketes

Hinweis Dieses Kapitel betrifft keine Snapshot basierte SW-Pakete. Diese werden wir bisher erstellt und verarbeitet.

Im Zuge der allgemeinen Tendenz zur Vereinfachung von Installationsaufgaben, werden auch SW-Pakete für Columbus mit einer übersichtlichen und nachvollziehbaren Verarbeitung erstellt. Diese Ausrichtung empfiehlt sich insbesondere beim Einsatz von Shareless für SW-Deployment.

Ziel ist es, Maschinen bezogene Aktionen ausschliesslich in der Server Sektion eines SW-Paketes zu verarbeiten statt wie bisher in Server und Client Sektionen.

Dies hat den Vorteil, dass nach der Verarbeitung einer Server Sektion, die verarbeitete Software effektiv zur Verfügung steht. Damit ist auch sichergestellt, dass Voraussetzungen mit einer tieferen *Identifizier* Reihenfolge vor einer Applikationsinstallation durchgeführt werden.

Die Client Sektion dient damit allenfalls für Aufräumarbeiten.

Benutzerspezifische Aktionen werden wie bisher in der Benutzer Sektion verarbeitet.

1.4 SW-Pakete im Einsatz für Shareless

Wenn SW-Pakete für Shareless eingesetzt werden, ist vor allem darauf zu achten, das ein SW-Paket alles enthält was innerhalb der Scriptteile verwendet wird. Während der Verarbeitung durch Shareless sind keine Zugriffe auf Fremdsysteme oder zusätzliche UNC Pfade möglich.

Da das gesamte SW-Paket als „Container“ zuerst lokal hergestellt wird, entspricht zur Laufzeit %_PkgSource% = %_PkgCache%. Es macht zum Beispiel keinen Sinn, Daten von %_PkgSource% nach %_PkgCache% zu kopieren, da diese in Shareless quasi lokal ein zweites Mal kopiert und gespeichert werden.

1.5 Konventionen für SW-Pakete

1.5.1 Nummerierung (Identifizier)

Die Nummerierung der SW-Pakete dient einerseits der eindeutigen Identifikation und andererseits um eine Reihenfolge der Installation / Deinstallation vorzugeben. Die Nummerierung beginnt bei 100000 und endet mit 999999. Somit ist gewährleistet, dass neue oder spezielle SW-Pakete vor oder nach den Bestehenden ausgeführt werden können.

Grundsätzlich werden innerhalb der Nummernblöcke 50er Schritte verwendet um neue SW-Pakete zu erstellen. Verwandte Applikationen können aber auch in 10er Schritte zusammengefasst werden.

Nummerblock	Beschreibung
100000 – 299999	Basis Konfiguration, Systemutilities, Treiber, Runtimes, Prerequisites, Player z.B. VMTools, ResourceKit, MS .net, C++, Java, Mediaplayer etc.
300000 – 499999	Standard Applikationen z.B. Microsoft, Adobe, SAP etc.
500000 – 899999	Restliche Applikationen z.B. Skype, Mozilla, VMware etc.
900000 – 949999	Brainware Produkte z.B. Packaging Tools, Console etc.
950000 – 999999	Abschlussarbeiten z.B. Final Config Package

Für jedes SW-Paket wird ein Identifizier zentral verwaltet um keine Dubletten oder zu ähnliche Identifizier zu erhalten.

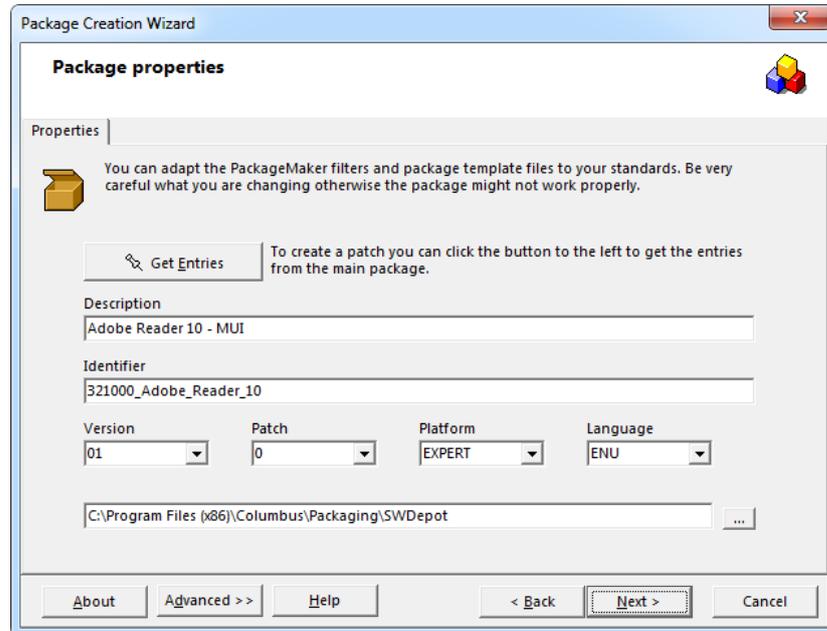
- Grundsätzlich gilt:
 NUMMER_HERSTELLER_PRODUKT_VERSION
- Es kann auch eine inhaltliche Angabe mitgegeben werden:
 NUMMER_HERSTELLER_PRODUKT_VERSION_INHALT
- Bei bekannten Herstellern kann mit einer Abkürzung gearbeitet werden:
 Microsoft MS
 Brainware BW
- Die Version wird mit der Hauptversion angegeben:
 7.03.09 7
 10.0.0.1 10

Beispiele:

Applikation	Identifizier
Adobe Reader 10	321000_Adobe_Reader_10
Microsoft Office 2007	301100_MS_Office_2007_Enterprise
Microsoft Office 2007 Settings Package	301100_MS_Office_2007_Enterprise_Settings
Microsoft Office 2003 MUI Package	386100_MS_Office_2003_MUI_LangPack

- Die Kombination aus obigen Angaben ergibt den Identifizier eines SW-Paketes.

- Wenn in einem bestehenden SW-Paket der Sprachcode geändert wird, so interpretieren die Management Console wie auch der Management Client dieses SW-Paket als Neuzugang.
- Die Sprache ist kein fester Bestandteil des Identifiers, dient jedoch der technischen Identifikation eines SW-Paketes.
- Der Name des SW-Paketverzeichnisses beinhaltet den Identifier wie auch die Sprache.



Hinweis Die Länge des Identifiers darf 46 (resp. 50 minus 4 zur Reserve) Zeichen nicht überschreiten.

Identifier (zentral)

Für jedes SW-Paket wird ein Identifier zentral verwaltet um keine Dubletten oder zu ähnliche Identifier zu erhalten.

Grundsätzlich gilt:

NUMMER_HERSTELLER_PRODUKT_VERSION

Es kann auch eine inhaltliche Angabe mitgegeben werden:

NUMMER_HERSTELLER_PRODUKT_VERSION_INHALT

Bei bekannten Herstellern kann mit einer Abkürzung gearbeitet werden:

Microsoft MS

Brainware BW

Die Version wird mit der Hauptversion angegeben:

7.03.09 7

10.0.0.1 10

Beispiele von Package Identifier:

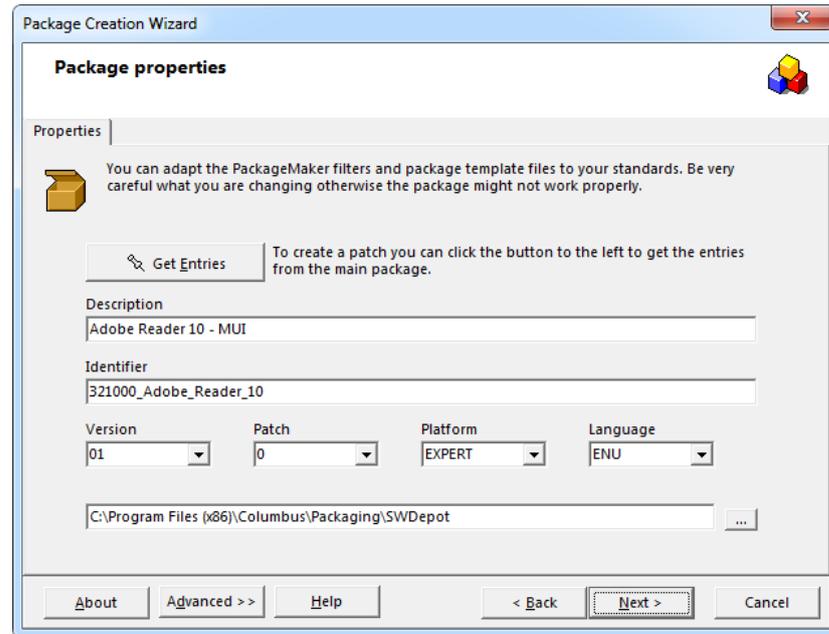
Applikation	Identifier
Adobe Reader 10	32100_Adobe_Reader_10
Microsoft Office 2007	30110_MS_Office_2007_Enterprise
Microsoft Office 2007 Settings Package	30110_MS_Office_2007_Enterprise_Settings
Microsoft Office 2003 MUI Package	38610_MS_Office_2003_MUI_LangPack

Die Kombination aus obigen Angaben ergibt den Identifier eines SW-Paketes.

Wenn in einem bestehenden SW-Paket der Sprachcode geändert wird, so interpretieren die Management Console wie auch der Management Client dieses SW-Paket als Neuzugang.

Die Sprache ist kein fester Bestandteil des Identifiers, dient jedoch der technischen Identifikation eines SW-Paketes.

Der Name des SW-Paketverzeichnisses beinhaltet den Identifier wie auch die Sprache.



Hinweis Die Länge des Identifiers darf 46 (resp. 50 minus 4 zur Reserve) Zeichen nicht überschreiten.

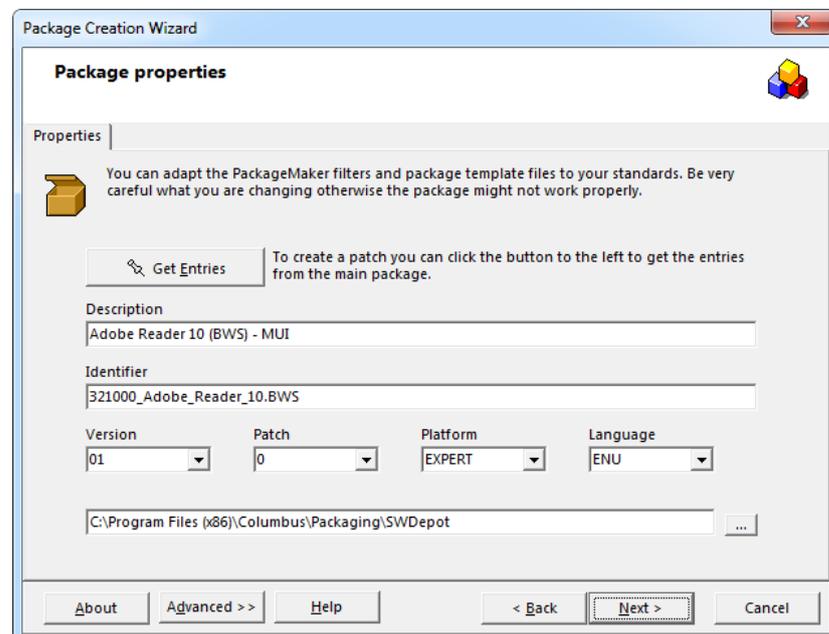
Identifier (Aussenstelle)

Für SW-Pakete der Aussenstellen wird im Identifier ein Kürzel integriert um die eindeutige Kennung sicherzustellen.

Eine Aussenstelle erstellt ein SW-Paket nur zur eigenen Verwendung:
32100_Adobe_Reader_10.nnn Identifier.Kürzel

Zu einem späteren Zeitpunkt benötigt eine zweite und dritte Aussenstelle dieses SW-Paket. Das SW-Paket kann nun gesamtheitlich analysiert und gegebenenfalls angepasst werden und wird danach in das zentrale SW-Release des Gesamtbetriebes aufgenommen.

321000_Adobe_Reader_10 Identifier gemäss Gesamtbetrieb (zentrales SW-Paket)



Hinweis Die Länge des Identifiers darf 46 (resp. 50 minus 4 zur Reserve) Zeichen nicht überschreiten.

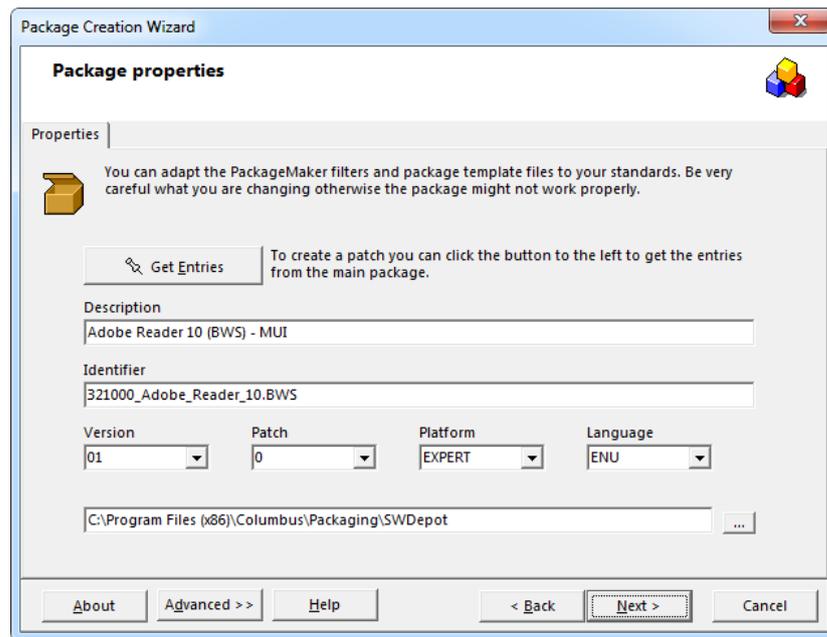
Identifier (zentral basierend)

Für zentrale SW-Pakete welche durch eine Aussenstelle geändert wurden, wird im Identifier ein Kürzel integriert um die eindeutige Kennung sicherzustellen.

Eine Aussenstelle verändert ein zentrales, bestehendes SW-Paket zur eigenen Verwendung:
 321000_Adobe_Reader_10 321000_Adobe_Reader_10.nnn

Somit ist die Herkunft des SW-Paketes klar ersichtlich. Bei Aktualisierungen des zentralen SW-Paketes, kann die Aussenstelle einfach feststellen ob diese Änderungen evtl. in das eigene SW-Paket übernommen werden müssen.

Änderungen an zentralen SW-Paketen haben keinen direkten Einfluss auf eigene angepasste.



Hinweis Die Länge des Identifiers darf 46 (resp. 50 minus 4 zur Reserve) Zeichen nicht überschreiten.

1.5.2 Description (zentral)

Name der paketierte Applikation. Bei der Zuweisung der SW-Pakete in der Management Console ist nur dieser Name sichtbar.

Grundsätzlich gilt: Hersteller Produkt Version - Sprache

Es kann auch eine inhaltliche Angabe mitgegeben werden: Hersteller Produkt Version Inhalt - Sprache

Bei bekannten Herstellern kann mit einer Abkürzung gearbeitet werden:

Microsoft MS
 Brainware BW

Die Version wird mit der Hauptversion angegeben:

7.03.09 7
 10.0.0.1 10

Beispiele von Package Description:

Applikation	Description
Adobe Reader 10 deutsch	Adobe Reader 10 - DEU
Microsoft Office 2007 englisch	MS Office 2007 - ENU

Microsoft Office 2003 Settings Package	MS Office 2007 Settings - UNI
Microsoft Office 2003 MUI Package	MS Office 2003 MUI LangPack - MUI

Hinweis Änderungen an der Description werden in der Management Console nach einem *Schedule Refresh Software Depot* des Software Deployment Agents, automatisch übernommen und lösen keine Neuinstallation des SW-Paketes aus.

1.5.3 Description (Aussenstelle)

Wenn innerhalb einer Organisation mehrere Administrationspunkte (IT-Abteilungen) vorhanden sind, welche eigene SW-Pakete erstellen sollen, so können diese mittels Kürzel voneinander unterschieden werden. Grundsätzlich gelten dieselben Konventionen wie vorher beschrieben, zusätzlich wird das Kürzel hinterlegt:

Hersteller Produkt Version - (nnn) - Sprache

- Adobe Reader 10 - (BWS) - DEU

1.5.4 Sprachcodes

Sprachcodes für SW-Pakete

Die Sprachcodes für SW-Pakete dienen innerhalb der *Description* oder des SW-Paketverzeichnisses zur grundsätzlichen Unterscheidung der Applikationssprachen.

Sprache	Code
Deutsch	DEU
Französisch	FRA
Italienisch	ITA
Englisch	ENU
Mehrsprachig	MUI
Universal	UNI

Sprachcodes im Benutzerteil

Die Sprachcodes für Benutzer dienen nur der sprachlichen Konfiguration von Applikationen innerhalb des **Benutzerteils** (UserAdd). Der Sprachcode kann **NICHT** zur regionalen Konfiguration / Identifikation genutzt werden.

Sprache	Code
Deutsch	DEU
Französisch	FRA
Italienisch	ITA
Englisch	ENU

Sprachcodes im Maschinenteil

Die Sprachcodes für Computer dienen nur der sprachlichen Konfiguration von Applikationen innerhalb der **Maschinenteile** (ServerAdd, ClientAdd). Der Sprachcode kann **NICHT** zur regionalen Konfiguration / Identifikation genutzt werden.

Sprache	Code
Deutsch	DEU

Französisch	FRA
Italienisch	ITA
Englisch	ENU

1.5.5 Version

Es hat sich bewährt, nur mit einer SW-Paket Version zu arbeiten: **01**

Die Funktion für weitere Versionen wird nicht mehr weitergeführt.

1.5.6 Patches

Haupt-SW-Paket muss immer die Patch-Version **0** aufweisen. Mit jedem Patch-SW-Paket wird diese Nummer erhöht.

Maximale Länge: **10 Zeichen**

Patches innerhalb von SW-Paketen können z.B. Updates, Upgrades oder eine Änderung an der Konfiguration der installierten Applikation beinhalten.

Zentrale Patches werden wie folgt nummeriert: 0100, 0200, 0300 usw.

Aussenstellen können eigene Patches hinzufügen oder ändern.

Zentrales SW-Paket	321000_Adobe_Reader_10
Zentraler Patch	01 0100
Aussenstellen Patch	01 0100nnn100

Zentrales SW-Paket	321000_Adobe_Reader_10	ohne Patch
Aussenstellen Patch	01 0000nnn100	

* nnn entspricht dem Firmenkürzel der jeweiligen Aussenstelle

1.5.7 Plattform (Komplexität)

Trägt zu einer eindeutigen Paket-ID bei, hat jedoch keine technische Funktionalität.

Maximale Länge: **10 Zeichen**

Die Plattform eines SW-Paketes dient zur Übersicht in der Management Console und hat keinen Einfluss auf die Lauffähigkeit eines SW-Paketes auf einem bestimmten Betriebssystem. Sie beschreibt die Komplexität eines SW-Paketes:

Plattform	Beschreibung
EASY	SW-Pakete mit simplem Aufbau ohne Verwendung von Konfigurationsdateien oder zentralen Variablen (Läuft ohne jegliche Aufwände auf jeder Columbus Umgebung)
ADVANCED	SW-Paket verwendet Konfigurationsdateien oder zentrale Variablen (Keine im SW-Paket fix enthaltenen Werte, alles über eine Konfigurationsdatei gesteuert)
EXPERT	Jeglicher Inhalt, Technologien, Variablen und Abhängigkeiten sind erlaubt / enthalten

Verwendungsmatrix:

Feature	EASY	ADVANCED	EXPERT
Voraussetzungen prüfen	X	X	X
Snapshot basiert			X
.msi, .exe (silent)	X	X	X

Feature	EASY	ADVANCED	EXPERT
Variablen in Console		(optional C_ConfigPath)	X
Konfigurationsdateien		X	X
Sprachumschaltung			X
Basis für Package2Go	X	X	
Bezug über PCC	X	X	X

1.5.8 Status

Der Status eines SW-Paketes zeigt die bereits abgeschlossenen, resp. die zukünftigen Arbeiten, welche am SW-Paket zu leisten sind.

Es werden folgende Status gesetzt:

Status	Beschreibung
Raw Snapshot	Neu erstelltes SW-Paket. Wird vom Package Engineer weiter bearbeitet.
Ready for QA	Für die QA bereit gestelltes SW-Paket.
Ready for Test	Für den Integrationstest vorbereitetes SW-Paket.
Ready for Pilot	Bereit um in einem Pilottest eingesetzt zu werden.
Productive	Das SW-Paket ist für den produktiven Einsatz freigegeben.
Obsolete	Das SW-Paket wird nicht mehr verwendet (keine Neu-Zuweisungen).
Archive	Das SW-Paket wird überall deinstalliert und danach aus dem SW-Release entfernt.

KAPITEL 2

SW-Pakete bearbeiten

In diesem Kapitel

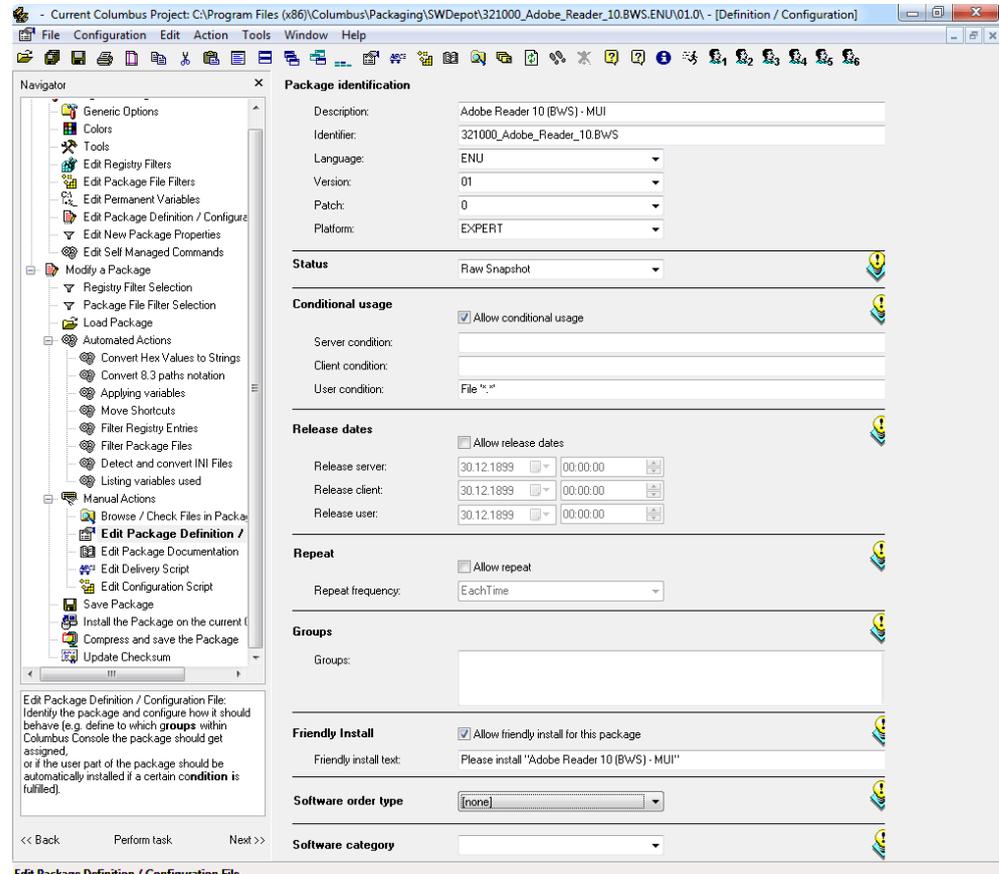
Package Definition / Configuration	18
Delivery-Script	20
Package Documentation	21
Configuration Script	22
Browse / Check Files in Package	23
Remove in SW-Paketen	23
SW-Paket Columbus Packaging Tools	24
Zusätzliche Werkzeuge in PackageStudio	24

Wichtig

Columbus SW-Pakete werden ausschliesslich mit Columbus PackageStudio bearbeitet.

2.1 Package Definition / Configuration

In dieser Übersicht wird die Definition des SW-Paketes, z.B. Description und Identifier gesetzt. Darüber hinaus kann das Verhalten des SW-Paketes auf einem Management Client gesteuert werden. Änderungen in dieser Übersicht bedürfen einem anschliessendem Neuimport in die Columbus Datenbank (Schedule Refresh Software Depot im Software Deployment Agent)

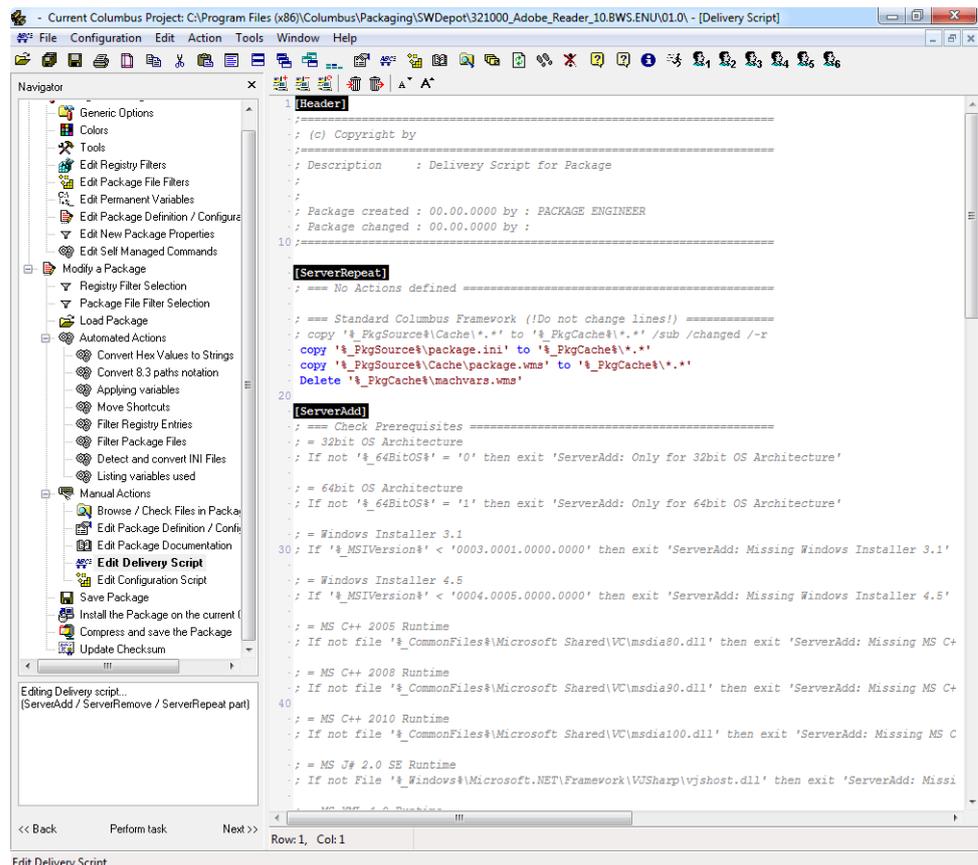


Abschnitt	Beschreibung
Package Identification	
• Description	Name der paketierte Applikation. Bei der Zuweisung der SW-Pakete in der Management Console ist nur dieser Name sichtbar.
• Identifier	Interner, alphanumerischer Name des SW-Paketes.
• Language	Information für die SW-Paketverwaltung.
• Version	Version des SW-Paketes (ausschliesslich 01)
• Patch	Haupt-Paket muss immer die Patch-Version 0 aufweisen. Mit jedem Patch-Paket wird diese Nummer erhöht.
• Platform	Komplexität des SW-Paketes
Status	Status des SW-Paketes im Qualitätsprozess
Conditional usage	
• Allow Conditional Usage	Definiert, ob das SW-Paket nur der Maschine zugewiesen werden muss und der User-Teil automatisch für jeden angemeldeten Benutzer ausgeführt wird. Wird standardmässig gesetzt um die Benutzerverwaltung zu vereinfachen.
• Server	Wird nicht verwendet

Abschnitt	Beschreibung
<ul style="list-style-type: none"> Client 	Wird nicht verwendet
<ul style="list-style-type: none"> User Condition 	File *.* Standard Einstellung, damit die automatisierte Benutzerzuweisung funktioniert
Release dates	Wird nicht verwendet.
Repeat	
<ul style="list-style-type: none"> Allow Repeat 	Definiert, ob das SW-Paket als Repeat eingetragen wird. Wenn ja, so wird bei jedem Start von Columbus der [ServerRepeat] und [ClientRepeat] Teil ausgeführt. Bei jedem Benutzerlogin wird die Sektion [UserRepeat] ausgeführt. Die Repeat Sektionen müssen manuell im SW-Paket erstellt werden.
<ul style="list-style-type: none"> Repeat Frequency 	Muss auf „Each Time“ gesetzt werden, damit obige Konfiguration aktiviert wird.
Groups	Mit Columbus 7 ist diese Einstellung direkt in der Management Console zu setzen.
Friendly Install	
<ul style="list-style-type: none"> Allow friendly Install for this Package 	Wenn auf einem Computer der Management Client so konfiguriert wurde, dass der Benutzer einer SW-Paket Installation zuvor zustimmen muss, so muss im SW-Paket die obige Konfiguration gesetzt werden. Ansonsten wird das SW-Paket unabhängig der Client Konfiguration installiert.
<ul style="list-style-type: none"> Friendly Install Text 	Dieser Text wird dem Benutzer bei der Aufforderung zur Installation angezeigt.
Software Order Type	Mit Columbus 7 ist diese Einstellung direkt in der Management Console zu setzen.
Software Category	Das SW-Paket kann in unterschiedliche Kategorien eingeteilt werden, damit ein Benutzer bei der Auswahl über den Software Kiosk daraus wählen kann.

2.2 Delivery-Script

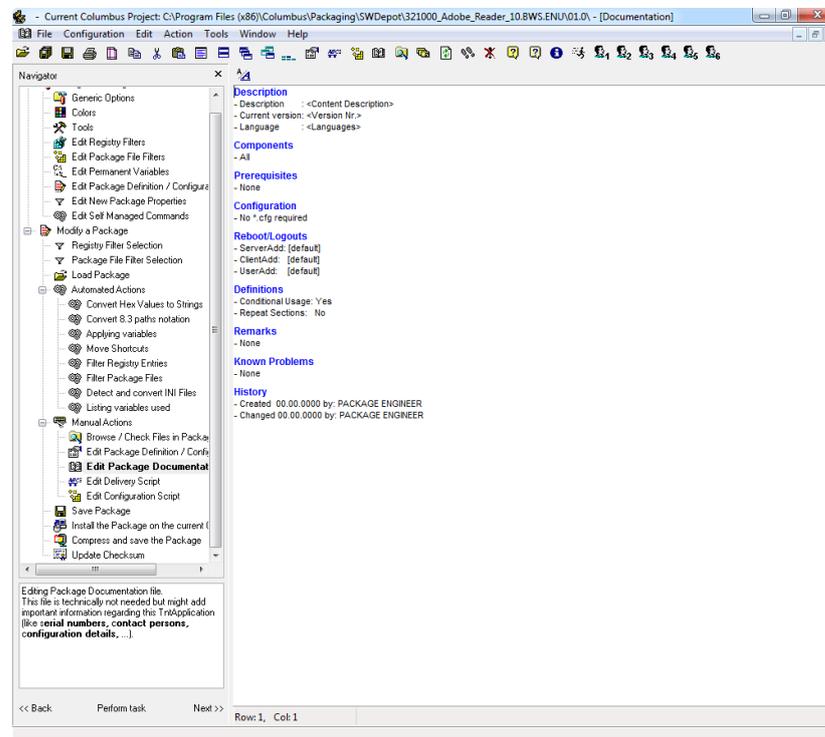
Das Delivery-Script beinhaltet die Serversektion (Teil der Maschineninstallation).



Sektion	Beschreibung
[Header]	Grundinformationen zum SW-Paket
[ServerAdd]	<p>Dieser Teil des SW-Paketes wird als erstes ausgeführt und beinhaltet meist eine Überprüfung auf vorinstallierte SW-Pakete oder Software.</p> <p>Wenn die allfälligen Prüfungen positiv abgeschlossen wurden, so werden Applikationsdaten und die weiteren Teile des SW-Paketes kopiert (Erweiterter Maschinenteil und der Benutzerteil)</p> <p>Weitere Aufgaben können sein:</p> <ul style="list-style-type: none"> • Installation und Reparatur • Änderungen an der Registry
[ServerRemove]	<p>Dieser Teil wird ausgeführt, wenn das SW-Paket in der Management Console auf <i>Remove</i> gesetzt wird.</p> <p>Aufgaben können sein:</p> <ul style="list-style-type: none"> • Deinstallation von Cleanup

2.3 Package Documentation

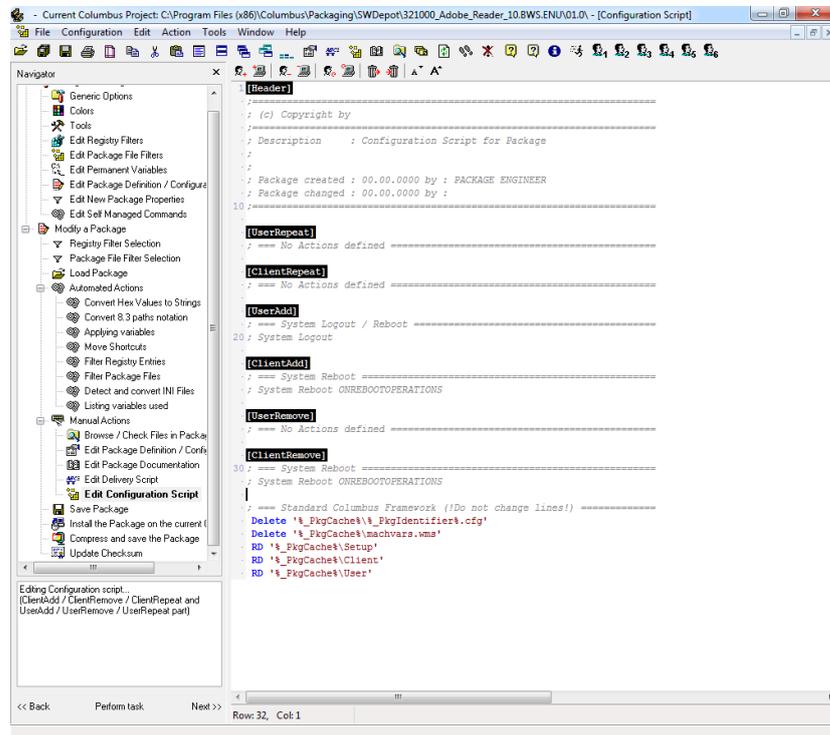
Die Angaben in dieser Übersicht können in der Management Console mit einem Doppelklick auf ein SW-Paket aufgerufen werden. Darin werden Angaben über den Inhalt des SW-Paketes gemacht. Die Angaben variieren je nach Detailgenauigkeit und Ansprüchen im Qualitätsprozess. Je genauer die dokumentierten Angaben sind, desto einfacher werden später Anpassungen oder Erweiterungen an SW-Paketen durchgeführt.



Sektion	Beschreibung
Description	Der Name des SW-Paketes, die aktuelle Version der installierten Software und die Sprache.
Components	Eine Applikation kann aus mehreren Komponenten bestehen, hier wird angegeben, welche davon installiert wurden. z.B. bei Microsoft Office können Word und Excel vorhanden sein, auf Outlook wurde verzichtet.
Prerequisites	Damit die zu installierende Applikation lauffähig ist, muss unter Umständen eine andere Applikation oder Runtime vorinstalliert sein, z.B. Microsoft .Net 1.1.
Configuration	SW-Pakete können mit Variablen dynamisiert werden. Die benötigten Variablen können in der Management Console oder auch in speziellen Konfigurationsdateien gespeichert sein.
Reboot	Vermerk, ob ein SW-Paket einen Systemneustart verlangt. Ein Systemneustart kann entweder gesammelt (System Reboot) oder direkt verlangt (System Reboot Immediate) werden.
Remarks	Bemerkungen zum SW-Paket (z.B. was zusätzlich zu beachten ist)
Known Problems	Bekannte Probleme, welche mit diesem SW-Paket auftreten können.
Creator	Ersteller des SW-Paketes. Sehr wichtiger Punkt, falls Probleme mit dem SW-Paket auftreten oder eine Änderung angestrebt wird.

2.4 Configuration Script

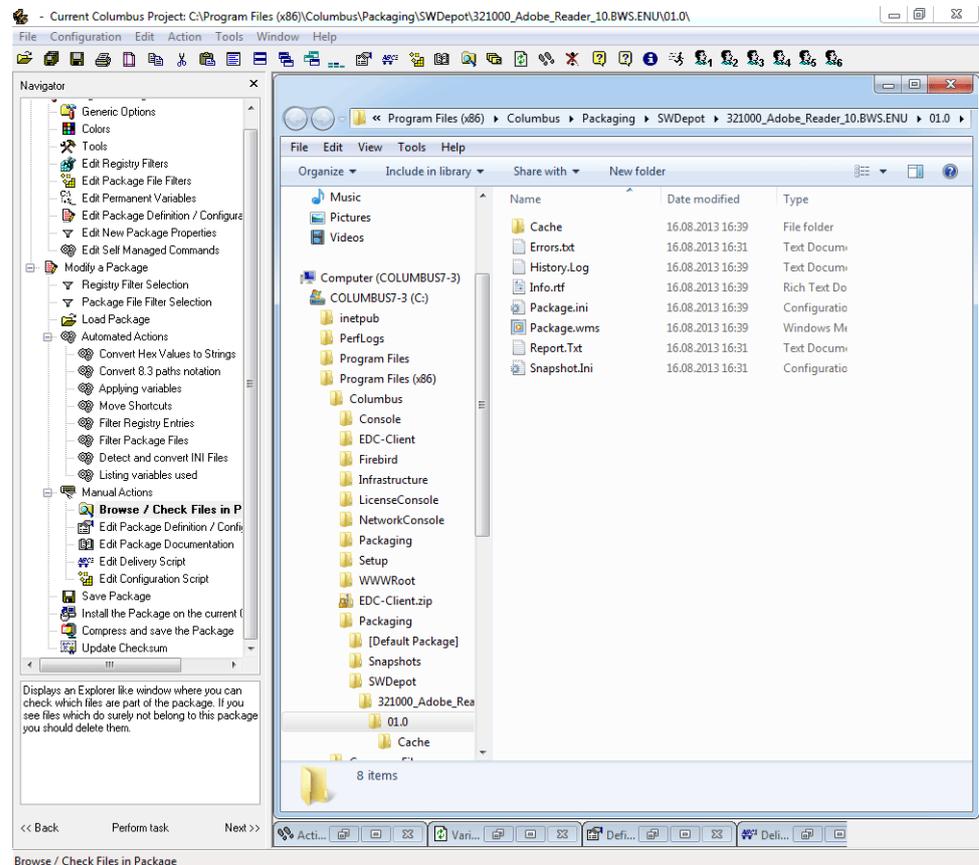
Das Configuration-Script beinhaltet den erweiterten Teil der Maschineninstallation sowie den Teil der Benutzerinstallation.



Sektion	Beschreibung
[UserAdd]	Ist für die Konfiguration / Installation der Applikation für den Benutzer zuständig. Die Scriptbefehle werden mit dem Kontext des momentan angemeldeten Benutzers ausgeführt. Weitere Aufgaben: <ul style="list-style-type: none"> • Application Data Einträge • User spezifische Variablen • Verknüpfungen im Benutzer Profil
[UserRemove]	Deinstallation der für den Benutzer relevanten Daten und Konfigurationseinstellungen. Weitere Aufgaben: <ul style="list-style-type: none"> • Entfernen von Benutzer-Variablen • Entfernen von Userkonfigurationen in Registry oder Konfigurationsdateien
[ClientAdd]	<ul style="list-style-type: none"> • Eventuell im ersten Maschinenteil nicht abgeschlossene Arbeiten
[ClientRemove]	Beinhaltet die Routine zur Entfernung der Applikation sowie das Aufräumen von Installationsdateien. Weitere Aufgaben: <ul style="list-style-type: none"> • Cleanup

2.5 Browse / Check Files in Package

In dieser Übersicht wird die Datenstruktur eines SW-Paketes angezeigt. Die Dateien und Verzeichnisse können bearbeitet oder gelöscht werden.



2.6 Remove in SW-Paketen

Unabhängig der Paketierungstechnik, sind folgende Regeln beim Remove zu beachten:

Task	Beschreibung
ServerRemove	<p>Im ServerRemove wird das Haupt-Applikationsverzeichnis gelöscht. Hier ist zu beachten, dass wenn Hersteller ins gleiche Hauptverzeichnis schreiben ist nur das Verzeichnis für die entsprechende Applikation zu entfernen.</p> <p>Im ClientRemove werden grundsätzlich keine Windows und System32 Dateien entfernt, ausser die Eindeutigkeit ist gewährleistet.</p> <p>Dateiverknüpfungen zur entfernten Software werden wieder frei gegeben.</p> <p>HKLM\Software Hauptschlüssel wird gelöscht. Hier gelten die gleichen Regeln wie beim ServerRemove</p> <p>AllUser Verknüpfungen werden entfernt</p>
ClientRemove	<p>Bis auf die vom Columbus Framework enthaltenen Scriptteile sind im Normalfall keine weiteren Aktionen notwendig.</p>
UserRemove	<p>Die User Verknüpfungen werden entfernt. Bei Verknüpfungsgruppen kann mittels If „%_Programs%\Adobe*.lnk" then RD „%_Programs%\Adobe" sichergestellt werden, dass das Verknüpfungsverzeichnis „Adobe" nur gelöscht wird, wenn darin keine Verknüpfungen mehr enthalten sind.</p>

Task	Beschreibung
	HKCU\Software Hauptschlüssel wird gelöscht Hier gelten die gleichen Regeln wie beim ServerRemove

2.7 SW-Paket Columbus Packaging Tools

Damit im produktiven Umfeld effizient und professionell SW-Pakete erstellt werden können, empfehlen wir PackageMaker und PackageStudio mit einem Columbus SW-Paket zu verteilen. Die **brainwaregroup** stellt hierzu ein SW-Paket zur Verfügung, welches jeweils die aktuellsten Paketierwerkzeuge und Dokumentationen beinhaltet.

Auswahl der benötigten Paketvorlage ([DefaultPackage])

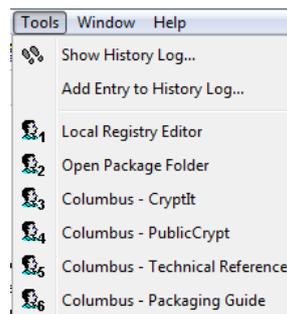
Im Speicherort der Paketierwerkzeuge (z.B. C:\Packaging) befinden sich mehrere Paketvorlagen. Diese sind gekennzeichnet mit [Default Package].#PLATFORM#. Wählen Sie je nach Bedarf die erforderliche Paketvorlage aus, in dem Sie das Verzeichnis manuell nach [Default Package] umbenennen (also das Suffix .#PLATFORM# entfernen).

Hinweis Wenn Sie keine Auswahl vornehmen wird die Paketvorlage für EASY verwendet.

2.8 Zusätzliche Werkzeuge in PackageStudio

Wenn die PackagingTools über das SW-Paket Columbus *PackagingTools 7 – UNI* auf einem Management Client installiert werden, wird der Funktionsumfang von PackageStudio mit Applikationen von Drittherstellern erweitert.

Über den Menüpunkt **Tools** werden folgende Werkzeuge angeboten:



Dieselben Werkzeuge werden bei einem geöffneten SW-Paket in der Funktionsleiste ebenfalls angezeigt:



Eintrag	Beschreibung
Local Registry Editor	„regedit“ der lokalen Maschine, mehrfacher Start möglich
Open Package Folder	Wenn ein SW-Paket geladen ist, wird der Pfad zu %_PkgSource% im Windows Explorer geöffnet
Columbus – Cryptit	Cryptit.exe um Passwörter im Gebrauch mit Columbus zu verschlüsseln
Columbus – PublicCrypt	PublicCrypt.exe um Passwörter im Gebrauch mit Drittherstellern zu verschlüsseln
Columbus – Technical Ref	Zugriff auf die <i>Technical Reference</i> von Columbus (Erklärung sämtlicher Columbus Scriptbefehle und weitere hilfreiche Informationen)
Project – Packaging Guide	Zugriff auf dieses Dokument

KAPITEL 3

Details zur SW-Paketierung

In diesem Kapitel

SW-Paketierungstechnik	25
Arbeiten mit „Packaging Tools“	25
Columbus Scriptsprache	26
Speicherort der Installationsquelle	26
Verknüpfungen	26
Konfiguration von SW-Paketen	26
Variablenauflösung - Columbus Scriptsprache	28
Informationen zu SW-Paketen	28
Repetitive Sektionen	29
Platzhalterpakete	29

3.1 SW-Paketierungstechnik

Wenn MSI oder Silent Setuproutinen vom Hersteller bereitgestellt werden, so werden diese mit den Basisangaben in ein Columbus SW-Paket übernommen und ausgeführt.

Oftmals ist eine Paketierung mit der Columbus Snapshot Technik einfacher zu bewerkstelligen, besonders in den Fällen, wenn die MSI oder Silent Setuproutinen gewisse Konfigurationenpunkte gar nicht enthalten oder die Installationsroutine nicht unattended / silent ausführbar ist.

Konfigurationen von Applikationen werden mit der Columbus Snapshot Technik erstellt und mitgegeben.

Die SW-Pakete werden mit einer firmenweit gültigen Packaging Vorlage erstellt.

Die Vorlage befindet sich innerhalb der *Packaging Tools* unter [Default Package]

3.2 Arbeiten mit „Packaging Tools“

Der Begriff *Packaging Tools* beinhaltet die für die SW-Paket Herstellung benötigten Programme und Hilfsmittel. In der Projekt Umgebung erhält jeder Betrieb ein SW-Paket, welches eine vordefinierte SW-Paketvorlage beinhaltet.

Das SW-Paket *902000_Columbus_PackagingTools.nnn.ENU* wird mit dem entsprechenden Kürzel komplettiert (Achtung: Identifier und Description) und die darin befindliche Paketvorlage wird falls nötig an die jeweilige Aussenstelle angepasst.

Dieses SW-Paket wird entgegen den übrigen geltenden Richtlinien jeweils in der Hauptversion verändert und an die Aussenstellen abgegeben (z.B. bei neuer PackageStudio Version).

3.3 Columbus Scriptsprache

Die Scriptsprache welche innerhalb von SW-Paketen (wie auch bei OSDeploy Jobs) ist eine mächtige aber dennoch übersichtliche Sprache. Die Grammatik der Columbus Scriptsprache erinnert an die bewährte Batchdatei Erstellung.

Die Befehle und Parameter werden laufend erweitert um den am Markt befindlichen Gegebenheiten zu entsprechen. Konsultieren Sie hierzu nach jedem Update der Columbus Umgebung die Command Help innerhalb von Columbus PackageStudio. Als Dokumentationsgrundlage dient das Handbuch Columbus **Technical Reference**.

Beachten Sie die korrekte Anwendung der Scriptsprache, damit die SW-Pakete fehlerfrei installiert werden können.

3.4 Speicherort der Installationsquelle

Grundsätzlich wird ein Paketverzeichnis als Gesamtpaket einer Applikation behandelt. Sämtliche notwendigen Dateien und Verzeichnisse sind darin enthalten.

Dies betrifft ebenso die Setuproutinen resp. die Installations-Sourcen. Damit ist gewährleistet, dass bei der Weitergabe der SW-Pakete, keine wichtigen Daten verloren oder vergessen gehen. Ebenso kann damit gewährleistet werden, dass die Installations-Sourcen nicht ohne entsprechenden Change Requests geändert werden.

Hinweis Installations-Sourcen werden nicht auf Columbus fremden Servern und Shares gelagert.

Innerhalb des SW-Paketes werden Installations-Sourcen unter folgendem Pfad abgelegt:

```
#IDENTIFIER#\01.0\Server\Setup\*.*
```

Falls eine Applikation nicht über das Netzwerk (Zugriff auf %_PkgSource%) installiert werden kann, so ist im Delivery-Script eine Alternative mit vorgängigem Kopiervorgang auf eine lokale Ablage enthalten.

```
Run '%_PkgSource%\Server\Setup\setup.exe' SHOW WAITDOWN TASK:5
```

3.5 Verknüpfungen

Grundsätzlich werden die Originalverknüpfungen der Setuproutine genutzt.

Wenn mit „Roaming Profiles“ gearbeitet wird (Servergespeicherte Benutzerprofile) werden die Verknüpfungen für Standard Applikationen im AllUser-Bereich abgelegt.

3.6 Konfiguration von SW-Paketen

Wenn in einem SW-Paket, Konfigurationen variabel (Standort-, Lizenzunabhängig) definiert werden müssen, so werden diese Angaben zusammengefasst pro SW-Paket in einer Konfigurationsdatei abgelegt.

Die Konfigurationsdatei wird benannt nach dem Identifier des dazugehörigen SW-Paketes. (z.B. 177500_Adobe_Reader_10.cfg oder auch 177500_Adobe_Reader_10.**BWS**.cfg)

Ebenso die Variablennamen (z.B. a_177500_Adobe_Acrobat_10_Serial oder auch a_177500_Adobe_Acrobat_10.**BWS**_Serial)

3.6.1 Namenskonvention von Variablen

Um eine Variable und deren Wert eindeutig einem SW-Paket zuweisen zu können, werden folgende Grundsatzunterschiede von Variablenamen definiert:

- _ Werden von Management Client verwaltet.
- a_ Wird innerhalb einer Konfigurationsdatei eines einzelnen SW-Paketes definiert.
- b_ Wird vom Basis Konfigurationspaket definiert
- c_ Wird innerhalb der Management Console definiert
- t_ Wird innerhalb eines SW-Paketes verwendet und in derselben Sektion wird entfernt.

Je nach Anwendungsbereich oder benötigter Überschreibungen von Werten, können ebenfalls a_ oder b_ Variablen innerhalb der Management Console definiert werden.

3.6.2 Aufbau einer Konfigurationsdatei

Wenn in einem SW-Paket, Konfigurationen variabel (Standort-, Lizenzunabhängig) definiert werden müssen, so werden diese Angaben zusammengefasst pro SW-Paket in einer Konfigurationsdatei abgelegt.

```
; =====  
; (c) Copyright by brainwaregroup  
; =====  
; Filename      : 177600_Adobe_Reader_10.cfg  
; Version       : 1.0  
; Change History : None  
; Created at    : 24.03.09 by : BWG/bdm  
; Changed at    : 00.00.00 by :  
; =====  
  
; === Client Settings =====  
[ClientSettings]  
a_177600_Adobe_Acrobat_10_Serial=1234-5678-9100  
  
; === User Settings =====  
[UserSettings]  
a_177600_Adobe_Acrobat_10_Update=0
```

In der Paketiervorlage für ADVANCED und EXPERT liegt ein Beispiel einer solchen Konfigurationsdatei unter #IDENTIFIER#\01.0\Config_#PKGIDENTIFIER#.cfg vor.

Die Konfigurationsdatei kann entweder auf einer zentralen oder dezentralen Ablage abgelegt werden.

Konfigurationsdateien werden nur für die Plattformen ADVANCED und EXPERT verwendet.

Zentrale Ablage

In Umgebungen mit mehreren Site Servern oder dem Bedürfnis einer zentralen Steuerung von SW-Paketten, kann mittels der Variable %C_ConfigPath% ein **zentraler Ablageort** (siehe "Variablenwerte über die Konsolen-Struktur" auf Seite 31) definiert werden.

Dezentrale Ablage

Die Konfiguration resp. Steuerung eines SW-Paketes kann auch dezentral erfolgen in dem die Konfigurationsdatei innerhalb des SW-Paketes abgelegt wird.

Das Beispiel einer Konfigurationsdatei wird wie folgt umbenannt und verwendet:
#PKGIDENTIFIER#.cfg > 410000_MS_Office_2010.cfg

3.6.3 Variable Werte in einem Paket

Um SW-Pakete Standort- und Lizenzunabhängig zu gestalten, werden einzelne Werte innerhalb eines SW-Paketes mit einer Variable ersetzt. In der Konfigurationsdatei werden diese Variablen mit den gültigen Werten abgefüllt.

Folgende Werte in SW-Paketen werden mit Variablen ersetzt:

- Lizenznehmer (Registered Organisation und Registered User)
- Lizenznummer
- Maschinentyp (Desktop, Notebook, Server)

3.7 Variablenuflösung - Columbus Scriptsprache

Der Columbus Script Interpreter löst Variablen von links nach rechts aus. Deshalb sind verschachtelte Variablen nicht möglich.

Kann aufgelöst werden

%_PkgIdentifizier%_License'	
Gelesene Werte von links nach rechts	Aufgelöst
%_PkgIdentifizier%	"200000"
_License	"_License"
Effektiv verarbeiteter Wert:	200000_License

Kann nicht aufgelöst werden

%a_%_PkgIdentifizier%_License%	
Gelesene Werte von links nach rechts	Aufgelöst
%a_%	" "
_PkgIdentifizier	"_PkgIdentifizier"
%_License%	" "
Effektiv verarbeiteter Wert:	_PkgIdentifizier

3.8 Informationen zu SW-Paketen

Im Moment der SW-Paketerstellung sind alle Angaben klar, die Netzwerkzugriffe funktionieren und die Reihenfolge sowie die benötigten Schritte zur Installation resp. Paketierung sind bekannt.

Wenn nun zu einem späteren Zeitpunkt an einem SW-Paket eine Änderung vorgenommen werden muss, fehlen häufig die obigen Angaben.

Innerhalb des SW-Paketes können unter Documents weitere Informationen hinterlegt werden.

Somit sind die Informationen auch für künftige Arbeiten gesichert vorhanden.

3.9 Repetitive Sektionen

Einzelne SW-Pakete können spezielle Sektionen enthalten, welche repetitiv ausgeführt werden.

Im Delivery-Script ist dies die ServerRepeat Sektion, im Configuration-Script sind dies UserRepeat und ClientRepeat Sektionen.

In diesen speziellen Sektionen wird dieselbe Columbus Scriptsprache verwendet wie für die übrigen Sektionen.

3.9.1 Ausführung der Sektionen

Die repetitiven Sektionen werden wie ihr jeweiliges Pendant ausgeführt:

- ServerRepeat wie ServerAdd (Maschinenteil)
- ClientRepeat wie ClientAdd (Maschinenteil)
- UserRepeat wie UserAdd (Benutzerteil)

Die repetitiven Sektionen werden zu jedem Zeitpunkt eines Columbus Prozessaufrufs durchgeführt (Push, Maschinenstart, Benutzerlogin etc.)

Während einem Benutzerlogin in einer Terminalserverumgebung wird nur die UserRepeat Sektion ausgeführt.

3.9.2 Herstellung der Sektionen

Wenn die Sektionen nicht im Template der "Packaging Tools" enthalten sind, so müssen diese manuell hergestellt werden.

In den Scriptteilen Delivery- und Configuration-Script, werden zwischen [Header] und erster [...Add] Sektion, die neuen repetitiven Sektionen erstellt.

3.9.3 Anwendungsbeispiele

Sektion [ServerRepeat]

```
; === Start RemoteRegistry Service for Support =====  
Service 'RemoteRegistry' /Start /NoErrors  
  
; === Change Password of local Administrator Account =====  
ChangePW '' '%_UserAdministrator%' 'H#1234ABCD'
```

Sektion [ClientRepeat]: Kein Handlungsbedarf

Sektion [UserRepeat]

```
; === Connect Networkdrive =====  
run 'net use w: \\SERVER\SHARE /PERSISTENT:NO' HIDE WAITDOWN TASK:5
```

3.10 Platzhalterpakete

In sehr seltenen Fällen wird die Applikation nicht vollständig paketiert, dies kann aus folgenden Gründen vorkommen:

- Applikation resp. der Setuproutine muss manuell aktiviert werden
- Es werden weniger als drei Installationen benötigt
- Die Paketierung würde unverhältnismässige Kosten verursachen

In solchen Fällen wird ein Platzhalterpaket erstellt, welches genau gleich wie die anderen SW-Pakete mit einem gültigen Identifier und einer Description erstellt werden.

Im SW-Paket enthalten kann die Setuproutine mit einer Installationsanleitung zur manuellen Installation.

Die Vorteile von Platzhalterpaketen:

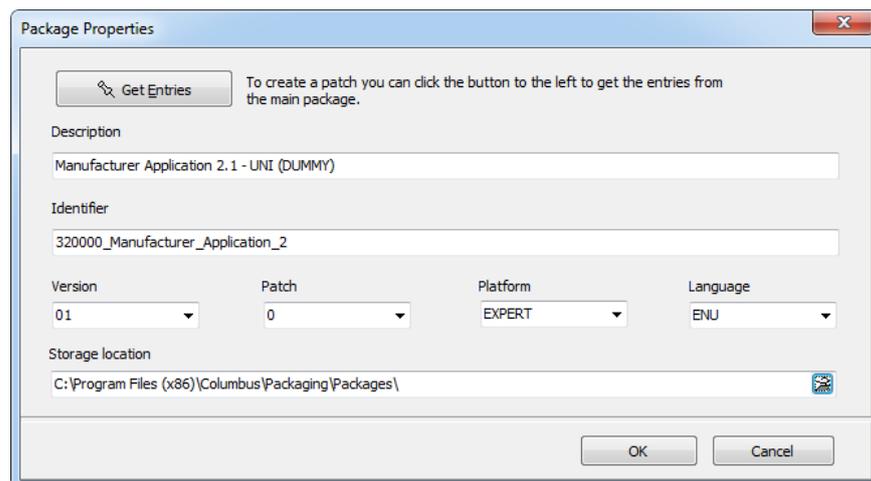
- Wenn später paketiert wird, besteht bereits die *Hülle*
- In der Management Console ist auf einen Blick erkennbar, welche Software auf dem Computer installiert werden sollte.

Um solche SW-Pakete von den übrigen zu unterscheiden wird in der Description der Wert (DUMMY) eingepflegt.

3.10.1 Namenskonventionen von Platzhalterpaketen

Es gelten die gleichen Konventionen wie bei Standardpaketen auch. Eine Unterscheidung wird nur bei der Description gemacht:

Hersteller Produkt Version – Sprache (DUMMY)



3.10.2 Vorgehen zur Erstellung eines Platzhalterpaketes

- Über die Funktion **Create empty Package** von PackageStudio kann ein neues SW-Paket erstellt und bearbeitet werden.
- Angaben unter **Edit Package Definition** anpassen.
- ... weitere Arbeiten wie gewohnt

KAPITEL 4

Variablenwerte über die Konsolen-Struktur

In diesem Kapitel

Information	31
Kurzübersicht.....	31
Detailerläuterung je Variable	32
Variablen pro Ebene	34
Variablenwerte vom Management Client	36

4.1 Information

Die Variablen, welche über die Management Console erstellt werden können, dienen der einfachen Verarbeitung von gleichen SW-Paketen aber unterschiedlicher Konfiguration.

Prinzipiell können die Variablen auf jeder denkbaren Ebene verändert werden. Grundsätzlich werden neue Variablen auf Company Ebene erstellt und bei der benötigten Ebene mit einem Wert versehen (immer vererbt, somit herrscht absolute Klarheit über deren Herkunft und deren Verfügbarkeit).

Die Anzahl der verwendeten Variablen in der Management Console sollte aus Gründen der Übersichtlichkeit auf rund 20 begrenzt sein. Konfigurationsangaben zu Applikationen (z.B. TwiXTel Server oder Exchange Server) werden nicht als Variable in der Management Console behandelt sondern als Konfigurationsdatei ausserhalb gespeichert.

Es werden nur Variablen gesetzt, wenn diese auch zur Anwendung kommen.

Hinweis Dieses Kapitel betrifft nur SW-Pakete mit der Komplexität ADVANCED oder EXPERT.

4.2 Kurzübersicht

Variable	Kurzbeschreibung	Mögliche Werte	Einsatz
c_ConfigPath	Pfad zu Konfigurationsdateien	\\Server\Share\Config \%c_Environment%	ADVANCED oder EXPERT wenn zentrale Ablage gewünscht.
c_MachineType	Maschinentyp	NOTEBOOK, WORKSTATION, SERVER	EXPERT
c_MachineLocation	Maschinen Standort	BERN	EXPERT
c_MachineLanguage	Maschinen Sprache	DEU, ITA, FRA, ENU, JPN	EXPERT
c_MachineDep	Maschinen Abteilung	Buchhaltung	EXPERT
c_UserDep	Benutzer Abteilung	Marketing	EXPERT
c_UserLanguage	Benutzersprache	DEU, ITA, FRA, ENU, JPN	EXPERT

4.3 **Detailerläuterung je Variable**

4.3.1 **c_ConfigPath**

Definition

- Pfad zu den Konfigurationsdateien, mit Rücksicht auf die Umgebung.

Verwendung

- Generell wenn zwischen Entwicklung und Produktion unterschieden werden muss
- Geografische Unterscheidung

Inhalt

- \\SERVER\Share\Config\%c_Environment% Ordner auf dem Haupt Columbus Server
- \\SERVER\Columbus\Config\%c_Environment% Beliebiger Servershare

4.3.2 **c_MachineType**

Definition

- Zur Unterscheidung von Maschinen Typen. Je nach Funktion einer Maschine.

Verwendung

- Wenn je nach Funktion unterschiedliche Konfigurationen erwünscht sind.
- z.B. Der Virens scanner soll bei Notebooks direkt auf die Hersteller Updatesite verbinden.
- z.B. Kein Reboot nach einer Installation auf Servern

Inhalt

- Workstation Fester Arbeitsplatz
- Notebook Mobiler Arbeitsplatz
- Server Server oder Serverfunktionalität
- Citrix Citrix oder Terminalserver

4.3.3 **c_MachineLocation**

Definition

- Standort der Maschine

Verwendung

- Wenn je nach Standort unterschiedliche Konfigurationen benötigt werden.
- z.B. Exchange Server in Bern oder Zürich

Inhalt

- BE Bern
- LBST09 Längenbühlstrasse 9
- HH Hochhaus
- Werkhalle 2 Zweite Werkhalle

4.3.4 **c_MachineLanguage**

Definition

- Primärsprache der Maschine

Verwendung

- Für Applikationen, welche nicht pro Benutzer unterschiedliche Spracheinstellungen zulassen
- Zur Einfacheren Handhabung von Einsprachigen Umgebungen, kann diese Variable auf Ebene *Company* gesetzt werden, somit müssen die Maschinen nicht in entsprechende Sites verschoben werden, sondern können allenfalls im allgemeinen Container *Computers* verbleiben.

Inhalt

- DEU Deutsch International
- ENU Englisch International
- FRA Französisch International
- ITA Italienisch International
- JPN Japanisch International

4.3.5 **c_MachineDep**

Definition

- Abteilung der Maschine

Verwendung

- Wenn je nach Abteilung unterschiedliche Konfigurationen benötigt werden.
- z.B. Verknüpfung auf Desktop bei allen „Werkstatt“ Maschinen.

Inhalt

- Werkstatt Maschine in der Abteilung Werkstatt
- Marketing Maschine in der Abteilung Marketing

4.3.6 **c_UserDep**

Definition

- Abteilung des Benutzers

Verwendung

- Wenn je nach Abteilung unterschiedliche Konfigurationen benötigt werden.
- z.B. Verknüpfung auf Desktop bei allen „Werkstatt“ Maschinen.

Inhalt

- Werkstatt Benutzer in der Abteilung Werkstatt
- Marketing Benutzer in der Abteilung Marketing

4.3.7 c_UserLanguage

Definition

- Primärsprache des Benutzers

Verwendung

- Für Applikationen, welche pro Benutzer unterschiedliche Spracheinstellungen zulassen
- Zur Einfacheren Handhabung von Einsprachigen Umgebungen, kann diese Variable auf Ebene *Company* gesetzt werden, somit müssen die Benutzer nicht in entsprechende Sites verschoben werden, sondern können allenfalls im allgemeinen Container *Users* verbleiben.

Inhalt

- DEU Deutsch International
- ENU Englisch International
- FRA Französisch International
- ITA Italienisch International
- JPN Japanisch International

4.4 Variablen pro Ebene

4.4.1 Auf Company Ebene

Variable	Mögliche Werte	Assign
c_ConfigPath	\\Server\Share\Config\%c_Environment%	Client
c_MachineTyp	{LEER}	Client
c_MachineLocation	{LEER}	Client
c_MachineLanguage	{LEER}*	Client
c_MachineDep	{LEER}	Client
c_UserDep	{LEER}	User
c_UserLanguage	{LEER}*	User

* Die Sprachvariablen können auch auf früheren Ebenen einheitlich gesetzt werden.

4.4.2 Auf Location Ebene

Variable	Mögliche Werte	Assign
c_ConfigPath	>> geerbt	Client
c_MachineTyp	{LEER} >> geerbt	Client
c_MachineLocation	BE, FR, ZG	Client
c_MachineLanguage	{LEER} >> geerbt *	Client
c_MachineDep	{LEER} >> geerbt	Client
c_UserDep	{LEER} >> geerbt	User
c_UserLanguage	{LEER} >> geerbt *	User

* Die Sprachvariablen können auch auf früheren Ebenen einheitlich gesetzt werden.

4.4.3 Auf Abteilung (Department) Ebene

Variable	Mögliche Werte	Assign
c_ConfigPath	>> geerbt	Client
c_MachineTyp	{LEER} >> geerbt	Client
c_MachineLocation	>> geerbt	Client
c_MachineLanguage	{LEER} >> geerbt *	Client
c_MachineDep	Buchhaltung, Support	Client
c_UserDep	Buchhaltung, Support	User
c_UserLanguage	{LEER} >> geerbt *	User

* Die Sprachvariablen können auch auf früheren Ebenen einheitlich gesetzt werden.

4.4.4 Auf Maschinentyp Ebene

Variable	Mögliche Werte	Assign
c_ConfigPath	>> geerbt	Client
c_MachineTyp	Workstation, Notebook, Server, Citrix	Client
c_MachineLocation	>> geerbt	Client
c_MachineLanguage	DEU, FRA, ITA, ENU, JPN *	Client
c_MachineDep	>> geerbt	Client
c_UserDep	>> geerbt	User
c_UserLanguage	{LEER} >> geerbt *	User

* Die Sprachvariablen können auch auf früheren Ebenen einheitlich gesetzt werden.

4.4.5 Auf Usertyp Ebene

Variable	Mögliche Werte	Assign
c_ConfigPath	>> geerbt	Client
c_MachineTyp	{LEER} >> geerbt	Client
c_MachineLocation	>> geerbt	Client
c_MachineLanguage	{LEER} >> geerbt	Client
c_MachineDep	>> geerbt	Client
c_UserDep	>> geerbt	User
c_UserLanguage	DEU, FRA, ITA, ENU, JPN *	User

* Die Sprachvariablen können auch auf früheren Ebenen einheitlich gesetzt werden.

4.5 Variablenwerte vom Management Client

Der eingesetzte Management Client bietet darüber hinaus weitere lokal geltende Variablen. Von der Version des Betriebssystems über das Vorhandensein von Windows-Komponenten bis zur Version der installierten Microsoft Office Suite. Sie finden die Variablen mit den dazugehörigen Werten in der Registry.

In allen Script-Teilen (Server, Client und User) anwendbare Variablen:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Brainware\Variables\Static
```

Nur im Benutzerteil (User) anwendbare Variablen:

```
HKEY_CURRENT_USER\SOFTWARE\Brainware\Variables\Static
```

KAPITEL 5

Log- und Konfigurationsdateien

In diesem Kapitel

Erstellung von Logdateien	37
Speicherort von Logdateien.....	37
Speicherort von Konfigurationsdateien	38
Spracheinstellung von SW-Paketen	38

Logdateien geben detaillierte Auskunft über Erfolg oder Misserfolg einer Setuproutine. Es wird grundsätzlich bei jeder Gelegenheit eine Logdatei erstellt und diese pro Computer zentral lokal gespeichert.

5.1 Erstellung von Logdateien

Wann immer möglich werden Logdateien geschrieben. Dies dient der Nachvollziehbarkeit bei Installations- oder Betriebsproblemen von Applikationen.

Damit die Logdatei eindeutig zu einem SW-Paket zugewiesen werden kann, wird die Variable `%_PkgIdentifizier%` verwendet. Somit ist gewährleistet, dass der jeweils gültige Identifizierer als Dateinamen verwendet wird:

```
"%_PkgCache%\%_PkgIdentifizier%-INSTALL.Log"
```

Insbesondere bei MSI Installationsroutinen wird eine Logdatei erstellt:

```
MSI '%_PkgSource%\Server\Setup\AcroPro.msi' 'ACTION="INSTALL" '  
/LOG:"%_PkgCache%\%_PkgIdentifizier%-INSTALL.Log" /NONE
```

Logdateien werden für jede MSI Setuproutine separat erstellt. Wenn in einem SW-Paket mehrere .msi ausgeführt werden, so wird für jede Setupinstanz eine eigene Logdatei erstellt.

```
MSI '%_PkgSource%\Server\Setup\AcroSP1.msi' 'ACTION="INSTALL" '  
/LOG:"%_PkgCache%\%_PkgIdentifizier%-SP1-INSTALL.Log" /NONE
```

Der Namen der Logdatei wird um die Instanz erweitert, entweder wird auf nummeriert, oder eine eindeutige Bezeichnung hinzugefügt.

Auch bei der SW-Paket Entfernung, wird eine Logdatei geschrieben und dementsprechend benannt.

```
MSI '%_PkgSource%\Server\Setup\AcroPro.msi' 'REMOVE="ALL" '  
/LOG:"%_PkgCache%\%_PkgIdentifizier%-UNINSTALL.Log" /NONE
```

5.2 Speicherort von Logdateien

Die Logdateien werden auf jedem Client lokal im Cache des jeweiligen SW-Paketes gespeichert. Der Speicherort wird mit der Variable `%_PkgCache%` definiert.

Die Logdateien verbleiben auf dem Computer und werden bei einer SW-Paket Entfernung nicht gelöscht. Damit kann jederzeit nachvollzogen werden, welche Setuproutine wann und wie durchgeführt wurde.

5.3 Speicherort von Konfigurationsdateien

In der Paketiervorlage für ADVANCED und EXPERT liegt ein Beispiel einer solchen Konfigurationsdatei unter #IDENTIFIER#\01.0\Config_#PKGIDENTIFIER#.cfg vor.

Die Konfigurationsdatei kann entweder auf einer zentralen oder dezentralen Ablage abgelegt werden.

Konfigurationsdateien werden nur für die Plattformen ADVANCED und EXPERT verwendet.

Zentrale Ablage

In Umgebungen mit mehreren Site-Servern oder dem Bedürfnis einer zentralen Steuerung von SW-Paketen, kann mittels der Variable %C_ConfigPath% ein **zentraler Ablageort** (siehe "**Variablenwerte über die Konsolen-Struktur**" auf Seite 31) definiert werden.

Dezentrale Ablage

Die Konfiguration resp. Steuerung eines SW-paketes kann auch dezentral erfolgen in dem die Konfigurationsdatei innerhalb des SW-Paketes abgelegt wird.

Das Beispiel einer Konfigurationsdatei wird wie folgt umbenannt und verwendet:

```
#PKGIDENTIFIER#.cfg > 41000_MS_Office_2010.cfg
```

Beispiel

```
; === Process Configfile =====  
; If file '%_PkgSource%\Config\_PkgIdentifizier%.cfg' then copy  
'%_PkgSource%\Config\_PkgIdentifizier%.cfg' to '%_PkgCache%\_PkgIdentifizier%.cfg'  
; If file '%c_ConfigPath%\_PkgIdentifizier%.cfg' then copy  
'%c_ConfigPath%\_PkgIdentifizier%.cfg' to '%_PkgCache%\_PkgIdentifizier%.cfg'  
; If not file '%_PkgCache%\_PkgIdentifizier%.cfg' then exit 'ServerAdd: Missing  
"%_PkgCache%\_PkgIdentifizier%.cfg"  
; Load Variables 'ClientSettings' '%_PkgCache%\_PkgIdentifizier%.cfg' /Volatile /Immediate  
; If '%a_51050_7-zip_Packager_9_Value%' = '' then exit 'ServerAdd: Missing  
%a_51050_7-zip_Packager_9_Value%'
```

5.4 Spracheinstellung von SW-Paketen

Die Sprache wird auf Benutzerebene über eine Variable definiert (%c_UserLanguage%)

Die Sprache gilt pro Benutzer. Mit dieser Konfiguration können unterschiedliche Benutzersprachen auf dem gleichen Computer verwendet werden.

Wenn eine Applikation nur einsprachig installierbar ist (Keine Multi Language Unterstützung), so wird über die MaschinenvARIABLE c_MachineLanguage die jeweilige Sprache gewählt.

Falls die Applikation nicht alle Firmensprachen unterstützt, so wird als Standardauswahl innerhalb eines SW-Paketes die Sprache Englisch gewählt.

Bsp. Italienisch ist nicht verfügbar:

```
If '%c_MachineLanguage%' = 'ENU' then run 'setup_english.exe'
```

KAPITEL 6

Software Abhängigkeiten

Um zu prüfen ob eine bestimmte Applikation vorhanden ist, wird im ServerAdd Teil eines SW-Paketes direkt die .exe Datei, eine .dll und / oder die Registry geprüft.

Es können auch die Static Variables von Columbus genutzt werden. Die detaillierten Variablen für .net oder andere Windows Komponenten sind nur mit einer PatchDeploy Lizenz verfügbar.

Beispiel

```
; = Windows Installer 4.5  
; If '%_MSIVersion%' < '0004.0005.0000.0000' then exit 'ServerAdd: Missing Windows Installer  
4.5'  
  
; = MS C++ 2005 Runtime
```

Hinweis

Um den SW-Release einheitlich zu gestalten und eine mögliche Fehlersuche zu vereinfachen, werden die Abfragen ausschliesslich in der Sektion ServerAdd durchgeführt.

Wichtig

In keinem Fall (ausser für das Base- und/oder Finalpackage) werden Abfragen erstellt, welche durch einzelne SW-Pakete mittels einer gesetzten Variable aufgelöst werden können.

Innerhalb einer derart grossflächigen und breit eingesetzten Plattform wäre es fatal, die Komponentenprüfung (z.B. Internet Explorer) über eine von einem SW-Paket gesetzte Variable zu erstellen, da ansonsten in jedem Fall dasjenige SW-Paket installiert werden müsste, welches diese Variable setzt obwohl die benötigte Komponente unter Umständen in den Windows Installationsdateien enthalten ist.

Dies wird umso schwerwiegender, wenn mit unterschiedlichen Betriebssystemversionen innerhalb einer Plattform gearbeitet wird (z.B. XP, 2003, Vista und 2008).

KAPITEL 7

Setuproutinen

In diesem Kapitel

Columbus eigener MSI Befehl	40
Mit Windows Installer	41
Windows Installer Fehlercodes	42
Weitere Setuproutinen	42
Reboot on demand	43

Der Windows Installer stellt eine Laufzeitumgebung für Installationsroutinen bereit.

Dieser Systemdienst kann Installationsroutinen in Form von MSI- und MSP (Patch) Dateien verarbeiten.

Neben Microsoft ist die Firma Installshield der grösste Anbieter von MSI Setuproutinen. Installshield arbeitet mit einer eigenen Scriptsprache und integriert diese in die MSI-Setuproutinen. Zur Ausführung von mit Installshield erstellten Setuproutinen, wird auf dem Zielcomputer ein InstallScript Framework benötigt.

Dieses InstallScript Framework ändert von Zeit zu Zeit und muss deshalb ebenfalls geprüft werden. Oftmals lassen sich Installshield Setuproutinen nicht installieren, weil das Framework der Zielmaschine veraltet oder defekt ist. Es wird empfohlen, die Setuproutine und das Framework zu prüfen und gegebenenfalls, das InstallScript Framework gesondert als SW-Paket zu Verteilen.

Wenn man den Marketingpapieren glauben mag, ist die Verarbeitung resp. die Verteilung von MSI basierenden SW-Paketen ein Kinderspiel. Tatsächlich müssen MSI Setuproutinen ebenfalls getestet, neu definiert oder komplett überarbeitet werden.

Die Konfiguration von Setuproutinen kann meist mit so genannten MST-Dateien bewerkstelligt werden, diese haben aber einige, zum Teil sehr brisante Nachteile:

- Nur mit zusätzlichem Arbeitsaufwand und einem Werkzeug erstellbar.
- Bietet nur die Möglichkeiten, welche vom Hersteller vorgesehen wurden.
- Können nachträglich nur schwer bearbeitet werden.
- Können nicht während der Installation verändert werden.
- Erhöhter Aufwand, wenn Benutzerspezifische Konfigurationen mitgegeben werden sollen.
- Konfigurieren nur die Setuproutine, nicht aber die installierte Applikation.

Es empfiehlt sich deshalb, die vom Hersteller freigegebene Setuproutine (MSI) direkt zu verwenden und mit Standard Parametern zu installieren. Anschliessend kann bequem mit der gewohnten Columbus Script-Routine, die gewünschte Konfiguration einer Applikation vorgenommen werden.

Somit können die praktischen MSI Installations- und, wenn verfügbar, die Reparaturroutinen, kombiniert mit der übersichtlichen Columbus Script Sprache zu einem sehr guten und umfangreichen SW-Paket verschmolzen werden.

7.1 Columbus eigener MSI Befehl

MSI Setuproutinen werden meistens mit dem Columbus eigenen MSI Befehl ausgeführt:

```
MSI '%_PkgSource%\Server\Setup\xyz.msi' 'REBOOT="ReallySuppress"  
/LOG:"%_PkgCache%\%_PkgIdentifizier%-INSTALL.log" /NONE
```

Viele MSI Setuproutinen verfügen über eine Reparaturfunktion, welche bei einer Re-Installation eines SW-Paketes genutzt werden kann.

Es wird nun zuerst geprüft, ob die MSI Setuproutine bereits installiert ist, und dementsprechend entweder komplett installiert oder nur die Reparaturfunktion genutzt:

```
If Installed 'MSIPRODUCT' '{74F6BA49-8345-44E6-A4E7-801497CC7C10}' then goto MSIReInst  
  
MSI '%_PkgSource%\Server\Setup\xyz.msi' 'REBOOT="ReallySuppress"  
/LOG:"%_PkgCache%\%_PkgIdentifizier%-INSTALL.log" /NONE  
  
goto EndMSICheck  
  
:MSIReInst  
MSI '%_PkgSource%\Server\Setup\xyz.msi' 'oumsv' /REPAIR  
/LOG:"%_PkgCache%\%_PkgIdentifizier%-REINSTALL.log" /NONE  
  
:EndMSICheck
```

Der Columbus eigene Befehl MSI, wertet die Errorcodes von MSI aus und informiert die Datenbank über Erfolg oder Misserfolg der MSI Installation. Falls die MSI Setuproutine nicht korrekt durchgeführt werden konnte, so wird der MSI eigene Fehlercode als Fehlerstatus in der Management Console angezeigt.

7.2 Mit Windows Installer

Wenn der Columbus eigene MSI Befehl nicht mit der Installationsroutine umgehen kann, so wird mittels direktem Aufruf des Windows Installers, die MSI Setuproutine ausgeführt.

```
Run 'msiexec /I "%_PkgSource%\Server\Setup\xyz.msi" REBOOT="ReallySuppress" /qn /L*v  
"%_PkgCache%\%_PkgIdentifizier%-INSTALL.log" Show Waitdown Task:30
```

Viele MSI Setuproutinen verfügen über eine Reparaturfunktion, welcher bei einer Re-Installation eines SW-Paketes genutzt werden kann.

Es wird nun zuerst geprüft, ob die MSI Setuproutine bereits installiert ist, und dementsprechend entweder komplett installiert oder nur die Reparaturfunktion genutzt:

```
If Installed 'MSIPRODUCT' '{74F6BA49-8345-44E6-A4E7-801497CC7C10}' then goto MSIReInst  
  
Run 'msiexec /I %_PkgSource%\Server\Setup\xyz.msi" REBOOT="ReallySuppress" /qn /L*v  
"%_PkgCache%\%_PkgIdentifizier%-INSTALL.log" Show Waitdown Task:30  
GOTO EndMSICheck  
  
:MSIReInst  
Run 'msiexec /Foums "%_PkgSource%\Server\Setup\xyz.msi" /qn /L*v  
"%_PkgCache%\%_PkgIdentifizier%-REINSTALL.log "' Show Waitdown Task:30  
  
:EndMSICheck
```

Mit obiger Methode wird nicht geprüft, ob die MSI Setuproutine erfolgreich durchgeführt wurde. Es kann also durchaus sein, dass das SW-Paket als erfolgreich abgeschlossen in der Management Console angezeigt wird, in Tat und Wahrheit aber die Applikation nicht installiert wurde.

Damit mit obiger Installationsmethode ebenfalls auf Fehler geprüft werden kann, wird nach Beendigung des Windows Installers durch einen Check des Rückgabewertes (ErrorLevel) der Status ermittelt:

```
Run 'msiexec /I "%_PkgSource%\Server\Setup\xyz.msi" REBOOT="ReallySuppress" /qn /L*v
"%_PkgCache%\%_PkgIdentifizier%-INSTALL.log ' Show Waitdown Task:30

If '%_ErrorLevelRun%' = '0' then goto NOERR
If '%_ErrorLevelRun%' = '3010' then goto NOERR
exit 'ServerAdd: Error "%_ErrorLevelRun%" occurred'
:NOERR
```

Dieser Check muss pro aufgerufener MSI Setuproutine in einem SW-Paket erstellt werden, dementsprechend ist die Sprungmarke NOERR aufzuzählen, um pro Check den richtigen Wert zu erhalten.

7.3 Windows Installer Fehlercodes

Die MSI Fehlercodes gelten sowohl für den Columbus MSI Befehl, wie auch der direkte Aufruf über den Windows Installer. Wenn die MSI Setuproutine ohne Fehler abgeschlossen werden konnte, so wird **0** zurückgemeldet.

Häufige Fehlercodes

Fehlercode	Beschreibung
0	Keine Fehler
3010	System Neustart erforderlich
1603	Unbekannter, fataler Fehler
1618	Es läuft bereits eine andere Installation
1619	MSI-Datei nicht aufrufbar

Informationen zu weiteren MSI Fehlercodes unter dieser Webadresse:
msdn.microsoft.com/en-us/library/aa368542.aspx

7.4 Weitere Setuproutinen

Neben der bekannten MSI-Setuproutine, sind noch weitere Setup-Hersteller und deren Produkte anzutreffen:

- Installshield
- InnoSetup
- Nullsoft
- Astarum

Auch diese Setuproutinen geben häufig einen Status in Form eines Rückgabewertes aus:

```
Run '%_PkgSource%\Server\Setup\setup.exe /s /1 %_PkgCache%\%_PkgIdentifizier%-INSTALL.log'  
Show Waitdown Task:30  
  
If '%_ErrorLevelRun%' = '0' then goto NOERR  
If '%_ErrorLevelRun%' = '3010' then goto NOERR  
exit 'ServerAdd: Error "%_ErrorLevelRun%" occurred'  
:NOERR
```

Die Aufrufe für Silent resp. Unattended Methoden variieren je nach Setup-Hersteller oder sogar innerhalb einer Setuproutinen Gemeinschaft.

Bekannte Parameter sind:

- /s auch /silent oder /verysilent
- /u auch /unattended
- /q auch /quiet

Hinweis

Gross- oder Kleinschreibung beachten.

Weitere Informationen zu Parametern sind entweder beim Applikations- oder Setuproutinen Hersteller erhältlich.

7.5 Reboot on demand

Ein Systemneustart der abhängig vom Returncode ist, kann folgendermassen realisiert werden:

```
Run '%_PkgCache%\Client\setup.exe /s /1%_PkgCache%\INSTALL.log' Show Waitdown Task:30  
  
If '%_ErrorLevelRun%' = '0' then goto End  
If '%_ErrorLevelRun%' = '3010' System Reboot  
If '%_ErrorLevelRun%' = '3010' then goto End  
exit 'ClientAdd: Error "%_ErrorLevelRun%" occurred'  
:End
```

KAPITEL 8

QA - Package Engineer

In diesem Kapitel

Allgemein	44
SW-Paket Eigentest - Vorgehen	45
SW-Paket Eigentest – Häufige Fehler	46

8.1 Allgemein

Inhalt der QA des einzelnen Package Engineer

Diese Vorgaben werden laufend erneuert und erweitert um den Package Engineer weitere Unterstützung zur einfachen Qualitätssicherung zu geben. Diese Vorgaben dienen als Checkliste, sind aber nicht abschliessend.

Namenskonvention

- Identifier gemäss Vorgaben erstellt und Sprachcodes eingehalten
- NUMMER_HERSTELLER_PRODUKT_VERSION
- NUMMER_HERSTELLER_PRODUKT_VERSION.KÜRZEL
- Sprachcodes der SW-Pakete eingehalten.
- Paketverzeichnis lautet IDENTIFIER.SPRACHE
- Hauptpaket trägt die Nummer 01.0
- Weitere Patches sind mit 0100, 0200 usw. aufgelistet

SW-Paket Aufbau

- Abfrage der Voraussetzungen in ServerAdd
- Entfernen der _PkgCache Daten in ClientRemove
- Keine fixen Angaben verwendet (Dafür Variablen gesetzt)
- Abhängigkeiten zu anderen SW-Paketen oder Applikationen beachtet

SW-Paket - Remove

- Remove der Applikation komplett.
- Logdateien bleiben erhalten (Rückschlüsse bei Problemen möglich)

Variablen

- Standard Variablen benutzt (%_Windows%, %_SystemDrive% usw.)
- Globale Variablen benutzt (%_Machinetyp%, %c_UserLanguage% usw.)
- Temporäre Variablen in der jeweiligen Sektion gelöscht (%t_ColData% usw.)
- Firmennamen und Benutzernamen mit Variable ersetzt (RegOwner, RegUser)
- 8.3 Variablen gesetzt

Scriptsprache

- Columbus Script Befehlssyntax eingehalten.

Inifiles oder ähnliche Konfigurationsdateien

- Inhalt auf fixe Verpfadungen geprüft und mit Textfile /Replace ersetzt

Verknüpfungen

- Verknüpfungen in UserAdd und ClientAdd geschrieben
- Verknüpfungen in UserRemove und ClientRemove entfernt
- Wenn Verknüpfungs-Gruppen entfernt werden, auf weitere vorhandene Verknüpfungen prüfen.
- Eventuell enthält der Verzeichnis weitere Verknüpfungen von ähnlichen Programmen

Dokumentation

- Info.rtf korrekt ausgefüllt (Description, Version, Angabe für *.cfg File, Prerequests angeben, regperm und xcacIs dokumentiert,
- *.cfg erstellt und ausgefüllt wenn benötigt
- SW-Paket Voraussetzungen aus ServerAdd in die Dokumentation übertragen
- Abfrage der unterstützten Plattformen (WORKSTATION, SERVER usw.) übertragen

Technisch

- Diverses

Funktionalität

- Funktioniert die Applikation mit eingeschränkten Benutzerrechten?
- Funktioniert die Applikation auf einer anderen Umgebung Hardware, Betriebssystem?
- Sind die definierten Einstellungen eingestellt?
- Treten Fehler in der Applikation auf?

8.2 SW-Paket Eigentest - Vorgehen

SW-Pakete, welche für die QA bereitgestellt werden, müssen bereits vorgängig getestet werden. Diese Tests beinhalten ebenfalls die korrekte Darstellung und Kommentierung der Columbus Scripts.

Der Absender eines SW-Paketes welches zur QA gelangt, ist verantwortlich für dessen Qualität. Dies auch, wenn ein SW-Paket vom Absender *nur* korrigiert wurde.

98% der SW-Pakete können ohne negative Nebeneffekte beim Testresultat, auf einer VM-Maschine getestet werden. Es gibt einige wenige SW-Pakete, welche HW abhängig sind, oder zusätzliche Peripheriegeräte (USB, Serial) benötigen.

Tests auf virtuellen Maschinen sind dank der VMware Snapshot Funktion sehr schnell umsetzbar. Von der Möglichkeit des Zurücksetzens sollte rege Gebrauch gemacht werden.

Trotzdem ist die VM-Maschine des Öfteren komplett neu zu installieren, um jeweils ebenfalls vom aktuellsten OS-Release oder den aktuellsten Basiskonfigurationen profitieren zu können.

Folgender Ablauf ist empfohlen:

Aufsetzen einer Testmaschine mit dem erforderlichen Betriebssystem

- MS Windows 7 SP1 english
- MS Windows 2008 SP1 english

Installation von Minimal Basispaketen

- (z.B. Base-Config, Windows MUI, Final-Config)

Zuweisen des SW-Paketes

- Werden fehlende Applikationen oder Komponenten angezeigt? (.net oder MS Office)

Hinzufügen von fehlenden Komponenten

- Fehlende Komponenten zuweisen (Eine nach der Anderen, dazwischen immer prüfen)

Maschine zurücksetzen, Benutzer wechseln (Standard User, keine Admin-rechte)

- Keine Benutzeranmeldung
- Warten auf die Statusmeldung in der Management Console.
- Anmeldung mit eingeschränktem Benutzer, prüfen der Ergebnisse

Re-Installation des SW-Paketes

- Mit angemeldetem Benutzer
- Ohne angemeldeten Benutzer

Zurücksetzen der Maschine, Ändern der Maschinensprache

- Gleiches Prozedere wie oben
- Am besten pro Sprache eine VMaschine bereitstellen

Zurücksetzen der Maschine, Ändern der Benutzersprache

- Gleiches Prozedere wie oben
- Am besten pro Sprache ein Benutzerkonto bereitstellen

Zurücksetzen der Maschine, Zuweisen der Standard SW-Gruppe

- Gleiches Prozedere wie oben
- Verträglichkeiten prüfen

8.3 SW-Paket Eigentest – Häufige Fehler

Obiger Testablauf ist ein Vorschlag, welcher je nach SW-Paket resp. der enthaltenen Applikation variieren kann. Je nach Erfahrung des Package Engineer können eigene Testroutinen zum Einsatz kommen.

Folgende Fehler und deren Konsequenzen werden häufig beobachtet:

Ein Computer für alles

- Tests auf dem gleichen Computer oder Snapshot
- Vorangehende Setupstörungen
- Unberechenbare Testergebnisse

Zuweisen der Standard SW-Gruppe (vor Testbeginn)

- Keine Kontrolle über mögliche, fehlende Komponenten

Test nur mit administrativem Account

- Keine Aussage gegenüber Endbenutzer

Test nur mit angemeldetem Benutzer

- SW-Pakete müssen ohne angemeldeten Benutzer installierbar sein.

KAPITEL 9

Info.rtf Vorlage

Description

This package installs:

- Description: <Content Description>
- Current version: <Version Nr.>
- Language: <Languages>

Components

- All

Prerequisites

- None

Configuration

- No *.cfg required

Reboot/Logouts

- ServerAdd: [default]
- ClientAdd: [default]
- UserAdd: [default]

Definitions

- Conditional Use: Yes
- Repeat Sections: No

Remarks

- None

Known Problems

- None

Creator

- Created 00.00.0000 by: PACKAGE ENGINEER
- Changed 00.00.0000 by: PACKAGE ENGINEER

KAPITEL 10

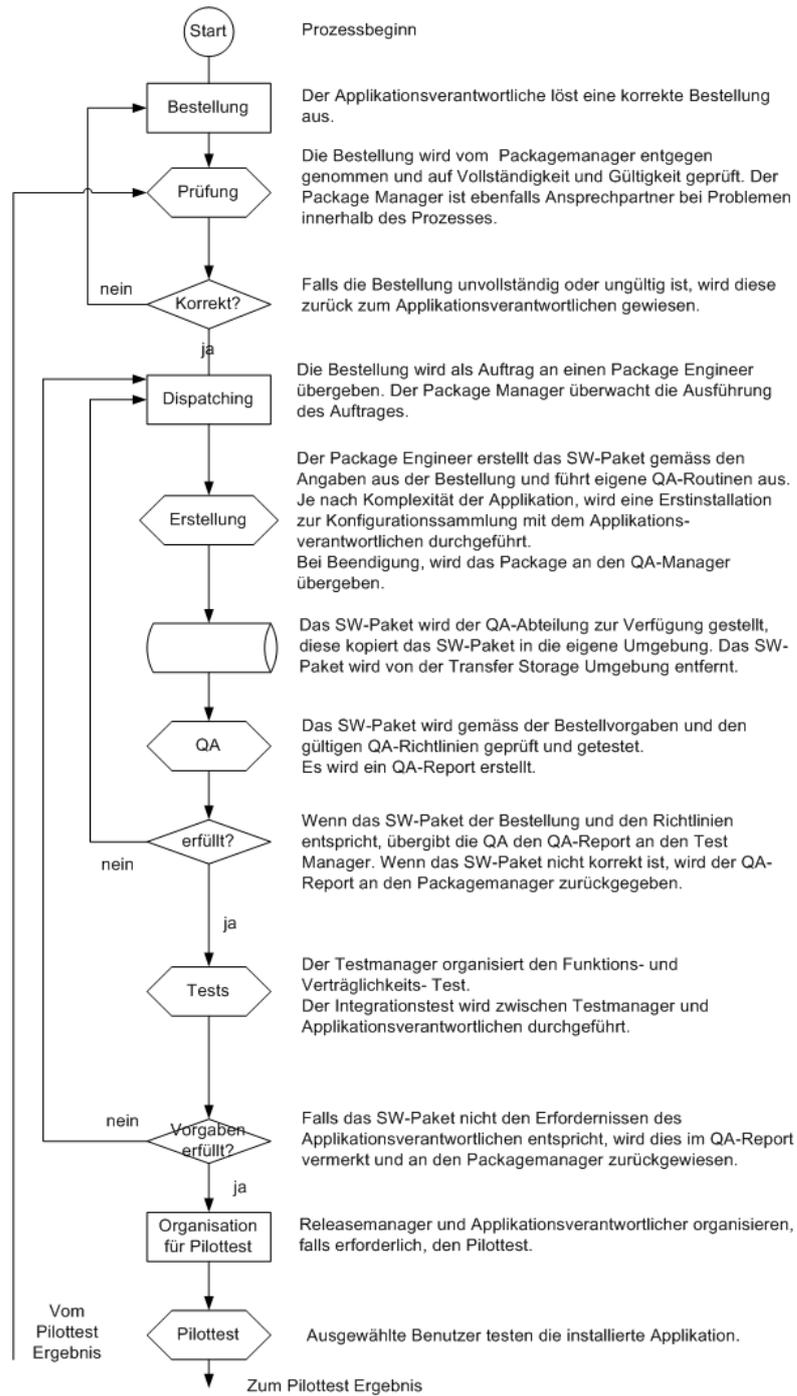
SW-Releasemanagement

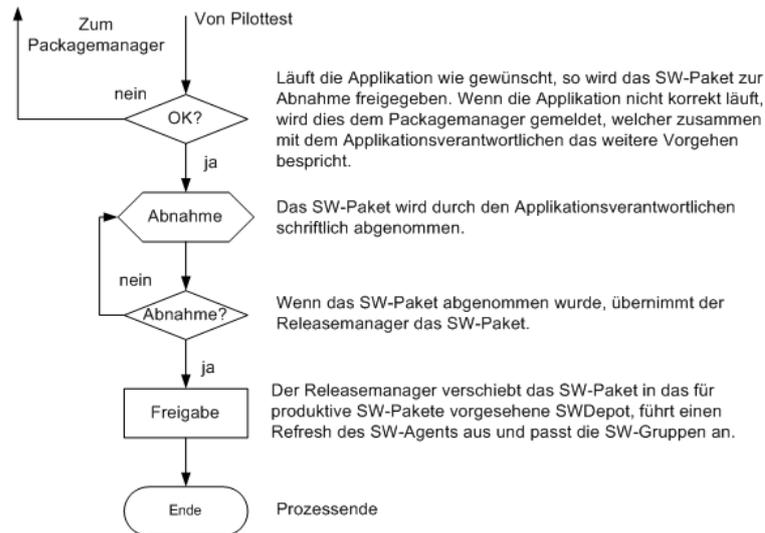
In diesem Kapitel

Prozessgrafik (Beispiel).....	50
Prozessrollen	51

Hinweis	Wird in einem separaten Dokument behandelt.
----------------	---

10.1 Prozessgrafik (Beispiel)





10.2 Prozessrollen

Prozessrolle	Beschreibung
Package Manager	<ul style="list-style-type: none"> • Erster Ansprechpartner bei Fragen oder Unregelmässigkeiten • Überwachung des Packaging Prozesses • Verteilt und überwacht die Packaging Aufträge • Ansprechpartner der Packaging Engineer
Applikationsverantwortlicher	<ul style="list-style-type: none"> • Erstellt die Vorgaben zur Erstellung des SW-Paketes (Den technischen Teil einer SW-Paket Bestellung) • Nimmt das SW-Paket schriftlich ab, bevor dieses vom Releasemanager in die produktive Umgebung überführt werden kann.
Releasemanager	<ul style="list-style-type: none"> • Überwacht den Stand der SW-Release • Definiert Datum und Zeit von Freigaben • Erhält sämtlichen Informationsfluss des Prozesses als Kopie
Package Engineer	<ul style="list-style-type: none"> • Erstellt SW-Pakete gemäss Bestellung und geltenden Richtlinien. • Macht Komponententests und Systemtests. • Arbeitet im 4-Augen Prinzip
QA-Manager	<ul style="list-style-type: none"> • Prüft und testet das SW-Paket, nicht aber die Applikation • Nimmt keine Änderungen an den SW-Paketen vor. • Erstellt den QA-Report • Gibt Empfehlungen • Rapportiert an Packagemanager.
Test-Manager	<ul style="list-style-type: none"> • Prüft und testet die Applikation • Nimmt keine Änderungen an den SW-Paketen vor. • Bearbeitet den QA-Report • Rapportiert an Packagemanager und Applikationsverantwortlichen
Pilottest Benutzer	<ul style="list-style-type: none"> • Ausgewählte Benutzer, welche die Applikation während des Pilottests ausführlich testen. • Rapportieren an int. QA-Manager und Applikationsverantwortlichen

KAPITEL 11

Glossar

In diesem Kapitel

Kürzel	52
Begriffe	52

11.1 Kürzel

Stelle	Kürzel
Geneva	GVA
Lausanne	LAU
Columbus Training	CTR
brainwaregroup	BWG

11.2 Begriffe

Begriff	Beschreibung
brainwaregroup	brainwaregroup ist der Hersteller von Columbus Softwaremanagement (SwM) und besteht aus den Teilen Entwicklung und Consulting. Die Produktreihe Columbus wird hauptsächlich in der Schweiz entwickelt. Weitere Informationen: www.brainwaregroup.com
Columbus	Oberbegriff der Softwaremanagement Applikation
Management Console	Steuerungseinheit des Infrastruktur Servers
Management Client	Installierte Instanz auf einem Computer. Zuständig zur Installation / Deinstallation, Inventur, Patch-Management usw.
Infrastruktur Server	Serverinstanz mit verschiedenen Agents und der Columbus Datenbank.
PackageMaker	Applikation um mit Vorher und Nachher Snapshots, SW-Pakete zu erstellen, welche danach in PackageStudio bearbeitet werden.
PackageStudio	Applikation mit welcher vorhandene Columbus SW-Pakete bearbeitet werden.
VMware	VMware resp. die Applikation VMware WorkStation bietet die Möglichkeit, Windows und andere Betriebssysteme auf einer virtuellen Maschine zu installieren und darauf SW-Pakete zu erstellen. Weitere Informationen: www.vmware.com (http://www.vmware.com)