

Einsatz und Bedienung des Conflict Explorers 2009

Inhaltsverzeichnis

1.	Einleitung	3
1.1	Konflikte in Softwarepaketen	3
1.2	Konfliktmanagement mit dem Conflict Explorer 2009	4
1.3	Überschaubare Investitionen und transparente Anpassungen, die die Qualität Ihrer Softwarepakete signifikant verbessern, zahlen sich aus!	4
1.4	Wie sinnvoll ist der Einsatz des Conflict Explorer 2009, wenn bereits andere Windows Installer-Entwicklertools mit Konfliktfeatures eingesetzt werden?	5
2.	Legende	6
3.	Terminologie	6
3.1	Familybase	6
3.2	Version	6
3.3	Package	6
3.4	Family oder Familienname	6
4.	Allgemeines.....	7
4.1	Features.....	7
4.2	Komponenten und Abhängigkeiten des Conflict Explorers.....	8
4.2.1	ImportMsi.EXE	8
4.2.2	CheckConflicts.EXE.....	10
4.2.3	ConflictExplorer.EXE	10
4.2.4	Settings.INI	10
5.	Bedienung des Conflict Explorers	14
5.1	Die Benutzeroberfläche	14
5.2	Importieren von Softwarepaketen.....	15
5.2.1	Importieren eines Softwarepakets über die Benutzeroberfläche	15
5.2.2	Automatischer Import mehrerer Softwarepakete in bestehenden Umgebungen	16
5.3	Grundsätzlicher Standard-Ablauf.....	17
5.4	Konfliktarten	17
5.4.1	Registry Conflicts:	18
5.4.2	SelfReg to SelfReg Conflicts	18
5.4.3	SelfReg to Registry Conflicts	18
5.4.4	Registry to SelfReg Conflicts	18
5.4.5	(Value Conflicts)	18
5.4.6	File Conflicts	19
5.5	Ausschluss von Konflikten	19
5.5.1	Filteroperationen	19
5.5.2	Organisatorische Massnahmen	19
5.5.3	Empfohlene Anwendung.....	21
5.6	Automatische Konfliktkorrektur.....	21
5.6.1	Besonderheit bei Registrierungsverweisen auf Dateien der Installationsinstanz.....	23
5.7	Manuelle Konfliktkorrektur	24
5.8	Anpassungen der ComponentId	24

5.9	Angewendete Standard-Regeln beim Ermitteln von Konflikten	27
5.10	Regeln beim Bilden des KeyPath bei neuen Komponenten	27
5.11	Suche	28
5.12	Optionen in der Benutzeroberfläche des Conflict Explorers	29
5.12.1	Register General	29
5.12.2	Register Resolve Settings	30
5.13	Konflikte exportieren	30
5.14	Pakete verschieben, kopieren und löschen, sowie Gruppen erstellen	31
5.14.1	Gruppe erstellen oder umbenennen	31
5.14.2	Paketfamilien verschieben oder kopieren	32
5.14.3	Paketfamilien löschen	33
5.15	Konflikte gegenüber dem Betriebssystem	33
5.16	Protokolldatei	34
5.17	Lizenz-Aktivierungsprozess	34
5.18	Kurztasten	36
5.19	Häufige Fragen	36
6.	Kommandozeilenoptionen	38
7.	Wichtige, bisher unbehandelte Funktionen	40
7.1	Delete Own Tasklist	40
7.3	Hilfe	40
8.	Stichwortverzeichnis	41

1. Einleitung

Erfolgreiches Software-Konfliktmanagement ist eine zentrale Aufgabe für eine robuste Software-Paketierung in einer grösseren Firmenumgebung. Ein Ignorieren dieser Aufgabe würde bedeuten, dass man die Augen vor den Folgen eines nicht erfolgten Konfliktmanagements verschliesst. Insbesondere nähme man die Risiken künftiger Störungen, Helpdesk-, sowie Supportinvestitionen und Nachbesserungsaufträgen in Kauf. Des Weiteren könnte das Image und Ansehen der Deploymentdienstleistung oder eines Migrationsprojektes stark darunter leiden, wenn man den Computerbenützer mit unnötigen Fehlersituationen auf seinem Arbeitsgerät konfrontiert.

1.1 Konflikte in Softwarepaketen

Die Begriffe „Konfliktbehebung“, „Konfliktvalidierung“ und „Konfliktlösung“ werden im Kontext der Softwarepaketierung und in verschiedenen Softwarelösungen unterschiedlich interpretiert: So finden wir bei einigen Werkzeugen Regelverstösse im Softwarepaket umfassenden und in sich abgeschlossenen lokalen Design als Ursache von Konflikten (*ICE*-Konflikte). In anderen Tools wird von Softwarekonflikten gesprochen, wenn eine bestimmte Datei aus einem Softwarepaket mit einer gleichbenannten Datei anderer Version aus einem anderen Softwarepaket in einer gemeinsamen Umgebung zur Installation kommt. Solche Tools erkennen diese Situation und schlagen beim Repaketierungsprozess oder bei nachfolgenden Konfliktmanagementoperationen ein Ersetzen der niedriger versionierten Datei im entsprechenden Softwarepaket vor. Und nicht zuletzt wird allgemein von Softwarekonflikten gesprochen, wenn im Softwarepaket COM-Komponenten verwendet werden, die sich über die Funktion *DllRegisterServer()* oder über anders implementierte Registrierungsmechanismen (bspl. *Registry*-Tabelle aus der MSI-Datei) bei der Installation registrieren. Werden solche COM-Komponenten schliesslich gemeinsam von verschiedenen Programmen verwendet, können durch das Überschreiben von bestehenden Registrierungswerten mit neuen Referenzen, dem Überschreiben bestehender COM-Komponenten mit nicht wirklich durchgängig kompatiblen Versionen dieser gemeinsamen Dateien oder dem Löschen von auf die DLL oder OCX bezogenen Registrierungsverweisen während der Deinstallation oder dem Upgrade eines Softwareproduktes Probleme entstehen, die ebenfalls unter dem Begriff „Softwarekonflikte“ zusammengefasst werden und allgemein als DLL-Hölle bekannt sind (DLL-Hell).

Natürlich gibt es noch viele weitere Ausprägungen von Softwarekonflikten in Softwarepaketen. Ihnen gemeinsam ist die Tatsache, dass sie sich erst im **späteren Verlauf**, in der gemeinsamen Verwendung verschiedener Softwareprodukte oder im Zusammenspiel von Softwareelementen mit dem Betriebssystem, sowie während oder nach Veränderung des Installationsstatus (Installation, Deinstallation, etc.) zu einem Problem manifestieren.

Die Folgen sind ganz unterschiedlicher Natur. Diese können im Einzelfall bedeuten, dass sich eine Software gar nicht erst erfolgreich installieren lässt, in anderen Fällen führen sie hingegen zum kompletten Versagen bestimmter Software oder auch nur zum Versagen bestimmter Funktionen, sowie zu fehlerhaften Auswirkungen nach Ausführung bestimmter Funktionen aus diesen Softwareprodukten. Dieses Verhalten lässt sich allgemein auf der gesamten Infrastruktur oder möglicherweise auch nur unter bestimmten Konstellationen im Unternehmen beobachten.

Wie wir nun gesehen haben, sind die Ausprägungen vielfältig und vielmals ist die nachträgliche Analyse, wie auch deren Konfliktidentifikation ein immens schwieriger und zeitaufwendiger Prozess, welche zudem das Verständnis komplexer Zusammenhänge erfordert. Dies ist auch der Grund, warum nicht selten die Ursache von Problemen im Verborgenen bleibt und man dem Thema „Softwarekonflikte in Softwarepaketen“ gelegentlich zu wenig Beachtung schenkt: Weil man vergangene Störfälle aufgrund von Softwarekonflikten nicht eindeutig als deren Folge erkannt hat. Umso wichtiger erscheint ein professionelles Software-Konfliktmanagement, welches ein **proaktives** Lösen von Problemsituationen anstrebt und die Auswirkungen von Softwarekonflikten bereits im Keim ersticken lässt.

1.2 Konfliktmanagement mit dem Conflict Explorer 2009

Wenn wir Softwarekonflikte aus Softwarepaketen kategorisieren, unterscheiden wir folgende Hauptgruppen:

1. Konflikte aufgrund von Regelverstößen beim Design eines Softwarepakets unter Berücksichtigung lokaler, meist auf das Softwarepaket beschränkter Regeln (z.B. *ICE*-Konflikte, *Internal Consistency Evaluators*)
2. Konflikte aufgrund globaler Designfehler und –konstellationen im Verbund mehrerer Softwarepakete oder im Zusammenspiel zwischen Software und Betriebssystem.

Der *Conflict Explorer 2009* ermittelt und löst Konflikte, die zu der zweiten beschriebenen Gruppe gehören. Nun kann man diese zweite Gruppe weiter unterteilen (folgende Aufzählung ist hierbei nicht abschliessend):

- Konflikte aufgrund gemeinsam verwendeter Ressourcen (Softwarepaket zu Softwarepaket oder Softwarepaket zu Betriebssystem), welche bei der Deinstallation oder beim Upgrade eines Softwarepakets entfernt werden und dadurch zu problematischen Situationen führen.
- Konflikte aufgrund gemeinsam verwendeten Ressourcen mit unterschiedlichem Inhalt (content), wie beispielsweise INI-Dateieinträge, die in verschiedenen Softwarepaketen unter dem gleichen Dateinamen identische Schlüssel (Key) verwenden, aber unterschiedliche Werte aufweisen. Auch in mehreren Softwarepaketen verwendete gleiche Registrierungsschlüssel mit unterschiedlichen Werten gehören zu dieser Gruppe.
- Konflikte aufgrund eines fehlerhaften Komponentendesigns, unter Berücksichtigung globaler Regeln (unterschiedlicher *KeyPath* bei „identischen“ Komponenten mit gleicher *ComponentId*, unterschiedliche Ressourcen „gleicher“ Komponenten, etc.)

Der *Conflict Explorer 2009* unterstützt das Konfliktmanagement von Konflikten aus all diesen drei Bereichen in der jetzt verfügbaren oder in zukünftigen Versionen. Genaue Angaben zu den zurzeit unterstützten Konfliktarten finden Sie unter [5.4 Konfliktarten](#).

1.3 Überschaubare Investitionen und transparente Anpassungen, die die Qualität Ihrer Softwarepakete signifikant verbessern, zahlen sich aus!

Bei einer Gesamtbetrachtung der Kosten innerhalb eines Applikations-Lebenszyklus lässt sich feststellen, dass qualitativ hochwertige und robuste Softwarepakete mit einem deutlich geringeren Gesamtaufwand zu Buche schlagen, als solche, die nicht diesen Standards entsprechen. Die Investitionen, die für den Mehrwert zu tätigen sind, stehen insbesondere mit dem Einsatz des *Conflict Explorers 2009* in keinem Verhältnis zu den Kosten, die ohne den Einsatz eines Software-Konfliktmanagements entstehen können. Denn ein Softwarebereitstellungsprozess ohne effektives Software-Konfliktmanagement kann letztlich schnell zum unkalkulierbaren Risiko heranwachsen.

In Gesprächen mit Vertretern aus der Softwarepaketierung trifft man hin und wieder auf die Aussage, man hätte bisher keine Probleme mit Softwarekonflikten gehabt. Tatsächlich bestätigt sich aber in den allermeisten dieser Fälle, dass man nur nichts direkt von den Auswirkungen von Softwarekonflikten wahrgenommen hat. In Szenarien, wo Computerbenutzer in einem Unternehmen ein Fehlverhalten in einer Software feststellen, versuchen diese zunächst oft selbst Hand anzulegen, um dem Fehler auf den Grund zu gehen. Dies ist ja an sich noch nichts verwerfliches. Bis die von Problemen geplagten Benutzer schliesslich zum Hörer greifen, um dem Unternehmenssupport anzurufen vergehen mitunter wertvolle Minuten bis Stunden. Bereits in dieser Phase entstehen einem Unternehmen nicht unerhebliche Kosten. Wenn auch der Unternehmenssupport nach einer Weile keine Lösung gefunden hat, wird in vielen Fällen eine Neuinstallation der betroffenen Software veranlasst oder beauftragt und nicht selten wird sogar das komplette System neu installiert. Über den Imageverlust der Informatikdienstleistungen gegenüber dem Kunden wollen wir hier mal gar nicht sprechen. Viel schlimmer wiegen die Kosten aller

Aufwände aller Beteiligten (Benutzer, Unternehmenssupport, für Softwareverteilung und Installationen zuständige Bereiche, etc.), die in der Summe eines Unternehmens (Konfliktsituationen entstehen in der Regel mehrfach) dramatische Grössen annehmen können. Das Besondere an der ganzen Geschichte ist, dass häufig das Team der Softwarepaketierung selbst gar nichts von diesen Auswirkungen mitbekommt, geschweige denn deren Ursachen kennt. Und für die Einzelfälle, wo man die Ursache dennoch klar lokalisieren konnte, wird schliesslich eine Nachkorrektur des betroffenen Softwarepakets unerlässlich; entstehen also wiederum neue Kosten.

Mit dem Einsatz des *Conflict Explorers 2009* wird nicht zuletzt die Tragweite bewusst, wenn man auf die integrierte, vornehmlich **proaktiv** ausgeführte, automatische Korrektur der Softwarekonflikte verzichten würde. In einem mittleren Unternehmen mit dreistelligem Softwarepool finden wir denn auch öfters Softwarepakete mit mehreren 1000 Einzelkonflikten zu anderen Softwarepaketen oder zum Betriebssystem. Erfahrungen, die wir in einem grösseren Unternehmen machen konnten, zeigten, dass dort unbehandelt etwa 25-30% aller Softwarepakete Softwarekonflikte aufwiesen. Dieser Wert ist natürlich abhängig von der Anzahl an importierten und verwendeten Softwarepaketen und steigt mit der Grösse des zum Einsatz kommenden Softwarepools an.

1.4 Wie sinnvoll ist der Einsatz des Conflict Explorer 2009, wenn bereits andere Windows Installer-Entwicklertools mit Konfliktfeatures eingesetzt werden?

Die meisten Windows Installer-Entwicklertools verwenden interne und lokale Konsistenzprüfungen (*Internal Consistency Evaluators, ICE*), um eine Paketvalidierung durchzuführen. Eine *ICE*-Validierung prüft die Datenbank des Pakets auf Einträge, die einzeln gültig sind, aber möglicherweise im Kontext der gesamten lokalen Datenbank ein falsches Verhalten verursachen könnten.

Der *Conflict Explorer 2009* erweitert die Prüfungsalgorithmen von **lokaler** Ebene auf die Unternehmensebene oder auf andere **globale** Sichten und ermöglicht die Darstellung von potentiellen Konflikten im heterogenen Desktopumfeld. Zudem ermöglicht der *Conflict Explorer 2009* eine automatisierte Konfliktlösung, die in Form einer robusten *Transformdatei* zum Windows Installer Paket hinzugefügt wird. Die Beschaffenheit dieser Anpassungen richtet sich nach Best-Practice Regeln und ist transparent und problemlos ohne weiteren Anpassungsbedarf für alle Arten von Windows Installer Paketen anwendbar.

Ausserdem unterscheidet er sich im Vergleich mit ähnlich gelagerten Tools in vielen Merkmalen, die sich zum Ziel gesetzt haben, den Software-Paketierer mit seinen Bedürfnissen im Zentrum zu sehen und die erfolgreiche Anwendung nicht zu Lasten der Softwarebereitstellungszeit umzusetzen.

Der *Conflict Explorer 2009* kann unabhängig verwendet werden. Er lässt sich aber ebenso elegant in bestehende Umgebungen in den Softwarebereitstellungsprozess integrieren. Zudem erweitert er allfällig bestehende Software-Konfliktmanagementlösungen mit professionellen Funktionen, die so bisher nirgendwo zu finden sind.

Weitere Features finden Sie unter [4.1 Features](#)

2. Legende

In *kursiv* geschriebene Wörter sind Namen, Fremdsprach- und Fachausdrücke, sowie Kapitelverweise. Auf Kapitelverweisen kann man mit „Ctrl. + Mausklick“ dem Link folgen

3. Terminologie

In der Terminologie des *Conflict Explorers* verwenden wir ein paar Ausdrücke mit ungewohnter Bedeutung.

Ein *Familyname* bezieht sich im Kontext des *Conflict Explorers* beispielsweise nicht auf die Produktfamilie (wie wir dies beispielsweise von der Gestaltung des *UpgradeCodes* her kennen), sondern auf eine *Paketfamilie*, mit all seinen Familienmitgliedern. Die Familienmitglieder werden durch ein oder mehrere *Packages* repräsentiert.

3.1 Familybase

Dies entspricht dem linken Teil des Paketnamen bis zu einem ersten allfällig verwendeten Trennzeichen „_“. Bspl. „ADOBEREADER“ (aus ADOBEREADER_9.0). Es ist nicht obligatorisch, in der Familienbezeichnung solche Trennzeichen zu verwenden. Die konsequente Verwendung dieses Trennzeichens ermöglicht es aber, Softwarepakete, die in der Produktion durch einen *Major Upgrade* mit einer neuen Version aktualisiert werden, weiterhin in der *Conflict Explorer* Datenbank zu belassen, ohne dass daraus gegenseitig Konflikte entstehen. Siehe auch [5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten](#).

Beispiel: ADOBEREADER_8.0 enthält dann keine Konflikte von ADOBEREADER_9.0 und umgekehrt.

3.2 Version

Dies entspricht der Versionsbezeichnung, bspl. „1.0“, welche optional im Familiennamen verwendet werden kann.

3.3 Package

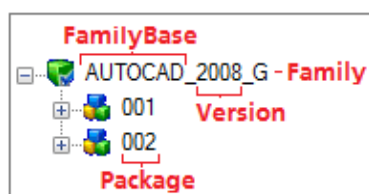
Ein *Package* ist ein eigenständiges MSI-Paket.

3.4 Family oder Familyname

Dies entspricht dem Namen der Software und ist unabhängig der *ProductName* Property. Der Name ist frei wählbar. Mit einer *Family* werden mehrere *Packages* zusammengefasst. Eine *Family* umfasst mindestens die *RTM* in Form eines *Packages*. Nachkorrekturen, *Small Updates* und *Minor Upgrades* können als weitere Bestandteile die *Family* erweitern.

In der Regel werden komplette Paketfamilien auf die Systeme installiert (ein *Package* nach dem anderen). Eine Transaktion in der Softwareverteilung (*INSTALL* oder *REMOVE*) bezieht sich folglich auf die gesamte *Family*. Das ist der Grund dafür, dass bei der Konfliktermittlung Konflikte unter *Packages* nicht berücksichtigt werden.

(siehe [5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten](#))



4. Allgemeines

4.1 Features

Der Einsatz des *Conflict Explorers* ist auf die Anzeige von Softwarekonflikten und die automatische Korrektur von Softwarepaketen ausgelegt. Folgende Features zeichnen ihn dabei vor allem im Vergleich zu Konkurrenzprodukten aus:

- Benutzergeführte und automatische Konfliktbereinigung in separate oder bestehende MST Datei.
- Selektive Korrekturmöglichkeit einzelner Konflikte nach Einstellungen des Benutzers.
- Problemlose Anwendung auch auf Hersteller-MSI-Dateien (nicht nur selbst repaketierte Pakete), ohne dass dadurch Folgeprobleme entstehen.
- Neben *Files*, *Registry*, etc. werden auch *SelfReg*-Informationen ermittelt und verglichen.
- Möglichkeit, Konflikte aus Paketen einer *Packagefamily-Excludeliste* mit einem Mausklick auszublenden.
- Möglichkeit, generelle, einzelne Konflikte via *Excludes* auszublenden.
- Echte Konflikte! (Eine Datei aus dem einen Paket gegenüber einer gleichbenannten Datei aus dem anderen Paket führt nicht automatisch zu einem Konflikt, wenn diese in unterschiedlichen Verzeichnissen vorgefunden wird.)
- Konsistente Daten. (Bessere Datenkonsistenz als Konkurrenzprodukte.)
- Arbeitserleichterungen.
(Sortieren von Konflikten, Spalten verschieben, Direktzugriff auf Einzelinformationen (copy in clipboard), etc.)
- Differenziertere und erweiterte Konfliktanzeigen (*Value Conflicts*, *Standard Conflicts*)
- Performanter und robuster Import! (Bisher sind keine MSI-Dateien bekannt, die nicht importiert werden konnten.)
- *Windows Installer Properties* werden grösstenteils aufgelöst, sofern sie nicht durch *CustomActions* entstehen. *AppSearch*-Properties werden in jedem Paket ermittelt und auch als Cleartext in die *Conflict Explorer* Datenbank übernommen.
- Unterstützung einer, mehrerer oder variabler MST-Dateien beim Import.
- Optische Sofortanzeige von Konfliktpaketen und Paketen, welche automatisch korrigiert wurden, im Treeview beim Browsen der Pakete (Icon). Das genaue Verhalten ist über die Benutzerkonfiguration einstellbar.
- Konflikte werden nicht nur zwischen Softwarepaketen ermittelt, sondern können auch zwischen Softwarepaketen und dem Betriebssystem dargestellt werden. So lassen sich gegenüber dem Betriebssystem konfliktfreie Softwarepakete realisieren.
- Auslesen und Anzeige des *Aktionsstatus* der Komponente, sowie Berücksichtigung des *Aktionsstatus* (*ActionState*) bei einer automatischen Konfliktbereinigung!
- Ansicht der Konflikte im *Listview* durch den Benutzer anpassbar (*display all Keys/Files just once*, *Display only Components which are installed*, etc.).
- Suche nach diversen Inhalten aus den Konflikten (F3).
- Praktische Anzeige zusätzlicher Informationen (Importdatum, Anzahl Konflikte vor/nach Operation, etc)
- Diverse praktische durch den Benutzer konfigurierbare Hilfen, zum einfachen und direkten Öffnen der zugrundeliegenden MSI-Datei mit einem Produkt nach Wahl (bspl. Orca), zum schnellen Navigieren in die Ablage der MSI (Explorer), etc. Berücksichtigung verschiedener Ablageverzeichnisse (Entwicklung/Test/Produktion, etc.)
- Erkennen von Abhängigkeiten über Konflikte.
- Einhaltung der Grundsatzregel: „Korrigiere soviel wie nötig, aber sowenig wie möglich!“

4.2 Komponenten und Abhängigkeiten des Conflict Explorers

Der *Conflict Explorer* wurde in .NET entwickelt und benötigt daher das .NET Framework 2.0, >= SP1 auf dem Client. Zusätzlich müssen die .NET Berechtigungen des Ausführungsverzeichnisses auf *FullTrust* geöffnet sein, damit der *Conflict Explorer* direkt ab einem Netzwerkshare gestartet werden kann. Diese Konfiguration übernimmt das Script *AddFullTrustingAndShortcuts.vbs* unter dem Verzeichnis *Client Setup*, welches auf jedem Arbeitsgerät auszuführen ist, wo der *Conflict Explorer* zum Einsatz kommt.

Hinweis:

Auf dem Arbeitsgerät sind zur Ausführung des Vorbereitungsscripts lokale Administratorrechte erforderlich. Unter Windows Vista erscheint ein entsprechender UAC-Prompt.

Der *Conflict Explorer* ist in drei verschiedene Module aufgeteilt. *ImportMsi.EXE*, *CheckConflicts.EXE* und *ConflictExplorer.EXE*. Zusätzlich befindet sich eine Konfigurationsdatei namens *Settings.INI* im Programmverzeichnis.

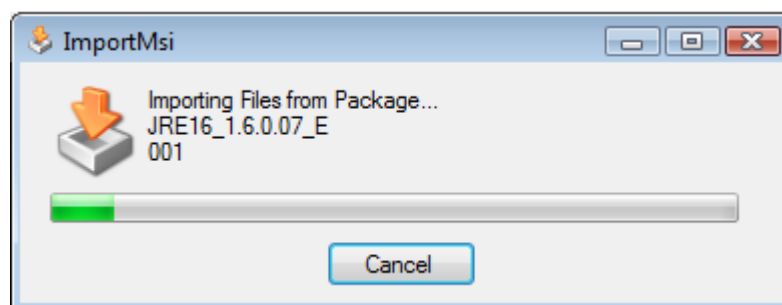
4.2.1 ImportMsi.EXE

Durch *ImportMsi* werden alle benötigten Informationen aus einem Windows Installer Paket in die *Conflict Explorer* Datenbank importiert. *ImportMsi.EXE* ist eine Konsolenanwendung und läuft auch ausserhalb der Benutzeroberfläche des *Conflict Explorers*. Dadurch lässt sich *ImportMsi.EXE* auch einfach in ein *Wise*-Projekt einbinden. Folgendes Bild demonstriert ein Beispiel-Projekt in *Wise Package Studio*:



Dahinter könnte folgender Aufruf stehen:

```
ImportMsi.EXE "[ProjectDir]\[FileName].msi" TRANSFORMS=ALL
```

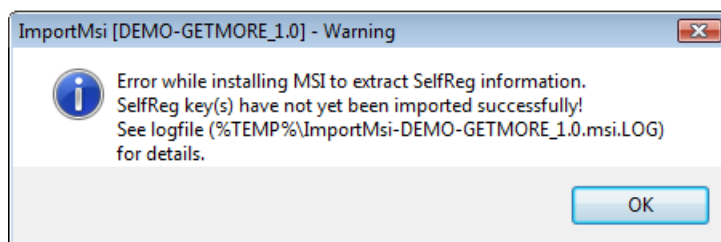


Werden innerhalb der MSI-Datei in der *SelfReg*-Tabelle Ressourcenverweise vorgefunden, so erfolgt durch *ImportMsi* eine Pseudoinstallation des Windows Installer Pakets, damit in der Folge aus PE-Dateien resultierende *SelfReg*-Inhalte ermittelt werden können. Nur so ist gewährleistet, dass auch Abhängigkeiten vorgefunden werden und die *SelfReg*-Registrykeys möglichst vollständig ermittelt werden können.

Hinweis:

Während dieser Pseudoinstallation können auch Fehler passieren. Oft sind solche Fehler darauf zurückzuführen, dass notwendige technische Abhängigkeiten fehlen (bspl. InstallScript Engine). In solchen Fällen kann man einfach per Hand die technische Abhängigkeit nachinstallieren und den Importprozess erneut starten.

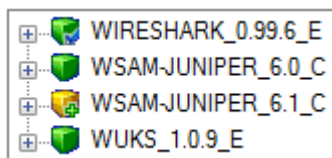
Aus Fehlern beim Importprozess resultierende mögliche Fehlermeldung:



Beachte:

Alle anderen Informationen werden unabhängig einer solchen Fehlermeldung dennoch in die *Conflict Explorer* Datenbank geschrieben.

Importierte, aber noch nicht auf Konflikte ermittelte (siehe folgendes Kapitel) Pakete werden im *Conflict Explorer* durch ein gelbes Icon in der *Package-Family* dargestellt.



Hinweis:

Während des Importprozesses kann auf viele Daten in der *Conflict Explorer* Datenbank nur lesend zugegriffen werden. Daher ist in einem Team auch nur ein Importprozess gleichzeitig durchführbar. Auf diese Situation wird in der Benutzeroberfläche des *Conflict Explorers* folgendermassen hingewiesen:

Users: oberlind (import) ▾

Hinweis:

Pakete sollten **nur auf einer Clean Machine** importiert werden, da sonst im Anschluss die *ActionStates* der verschiedenen Komponenten falsche Ergebnisse aufweisen könnten. Siehe auch Absatz *CleanMachinePackageLimitValue* im Kapitel 4.2.4 *Settings.INI*

ImportMsi kann aus dem *Conflict Explorer* über das Menü *Package/Import* gewählt oder mit oder ohne entsprechende Kommandozeilenoptionen auch direkt ausgeführt werden.

Hinweis:

Will man für die Namensgebung generell und automatisch die Namen aus der *Property*-Tabelle der MSI-Datei auslesen, so kann man über die *Settings.INI* unter *[Global]* die Einträge *PropertyPackage=PROPERTY* und *PropertyFamily=PROPERTY* hinzufügen. ‚PROPERTY‘ ist dabei mit dem gewünschten Eigenschaftsnamen zu ersetzen. Die entsprechende *Property* muss natürlich im Softwarebereitstellungsprozess vorgängig abgefüllt werden!

Beispiel: *PROPERTYFAMILY=ProductName*.

Alternativ ist es möglich, den *Familyname* und die *Package*-Bezeichnung über Kommandozeilenoptionen anzugeben:

`ImportMsi.EXE "\\MYSERVER\Package\Reader.msi" FAMILY=ADOBEREADER_9.0 PACKAGE=001`

Zusätzlich kann mit der Kommandozeilenoption **GROUPS='Name';'Name'** *ImportMsi* dazu veranlasst werden, dass das *Package* automatisch bestimmten *Gruppen* zugewiesen wird. Fehlt diese Kommandozeilenoption, dann wird das Paket der *Gruppe* ‚unassigned Packages‘ zugewiesen, woraus es später über die *Conflict Explorer* Benutzeroberfläche in die richtige *Gruppe* verschoben werden kann.

4.2.2 CheckConflicts.EXE

Mittels *CheckConflicts.EXE* werden Konflikthanzeigen erst ermöglicht. Hierbei handelt es sich um ein Modul, welches die Ressourcen eines Paketes mit den Ressourcen aller anderen Paketen vergleicht. Optional können auch alle Ressourcen aller Pakete mit allen Ressourcen aller Pakete verglichen werden. Dieser Prozess kann aber je nach Anzahl importierter Pakete, Anzahl *Excludes*, Anzahl *Gruppen* und Anzahl Ressourcen grössere Zeit in Anspruch nehmen. Im Paketierprozess, wenn sich die Vorhaben auf ein Paket beschränken, ist jeweils ein „*Scan Package Conflict*“ empfohlen, welcher ein Paket mit allen Paketen vergleicht. Dieser Prozess ist in Relation der Daten- und Prüfmengen sehr performant.



4.2.3 ConflictExplorer.EXE

Dies ist die Benutzeroberfläche des *Conflict Explorers*.

Weitere Informationen findet man unter dem Kapitel [5 Bedienung des Conflict Explorers](#)

4.2.4 Settings.INI




In die Datei *Settings.INI* werden globale Settings eingetragen, die für alle User gelten. Einige Eigenschaften werden hierbei beim Verändern durch den Benutzer zusätzlich in der Registry unter *HKCU\Software\Conflict Explorer* abgespeichert. Bei einigen Einstellungen gewinnen die Einträge des Benutzers vor den Einstellungen der *Settings.INI*, bei anderen hingegen, können Einträge aus der *Settings.INI* durch den Benutzer nicht überschrieben werden (wird bei der Einstellung dokumentiert).

OrcaPath

Hier kann ein MSI Anzeige- oder Editierprogramm deklariert werden. Auf diesen Pfad greift der *Conflict Explorer* beim Klick auf den Toolstrip-Button  zu.

```
[Global]
OrcaPath=\\MYSERVER\Integration\Tools\Orca3.1.4000\Orca.EXE
```

PathReplacements

Mit dieser Option ist es beim Öffnen der MSI über «Edit MSI»  oder beim Direktzugriff  auf den Dateipfad möglich, bei Nichtvorfinden der MSI-Datei die Suche automatisch auf alternative Ablagepfade auszudehnen. Auch ein *Relimport*  verwendet im Falle des Nichtvorfindens der Installationsdatei dieser Ersetzpfade. Wenn Ihre Softwarepakete im Softwarebereitstellungsprozess nacheinander auf verschiedene Server oder Verzeichnisse verschoben werden (max. 3), dann korrigieren Sie diesen Eintrag mit Ihren Abbildungen. Trennen Sie die Verzeichnisse mit Semikolon. Der erste Eintrag muss dem Standardverzeichnis entsprechen, woraus die Pakete importiert werden. Dieses wird bei Fehlen der Installationsdatei durch das zweite oder schliesslich das letzte Verzeichnis ersetzt. Beachten Sie, dass nur der linke veränderbare Teil des Verzeichnisses angegeben werden muss, wenn die darunterliegende Verzeichnisstruktur jeweils identisch aufgebaut ist.

```
[Global]
PathReplacements=\\MYSERVER\Development\Packages;\\MYSERVER\Integration\Packages;\\MY...
```

Beispiel:

Entwicklungsshare: [\\MYSERVER\Development\Packages](#)

Integration/Test: [\\MYSERVER\Integration\Packages](#)

Produktion: [\\MYSERVER\Production\Packages](#)

Beispiel-Softwarepaket wurde importiert aus:

[\\MYSERVER\Development\Packages\German\ADOBEREADER\001](#)

Beispiel-Softwarepaket befindet sich jetzt in:

[\\MYSERVER\Production\Packages\German\ADOBEREADER\001](#)

Mit folgendem Eintrag wird gewährleistet, dass der *Conflict Explorer* jederzeit auf die Installationsdatei zugreifen kann:

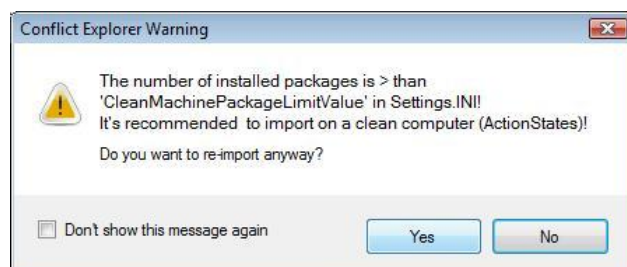
PathReplacements=\\MYSERVER\Development;\\MYSERVER\Integration;\\MYSERVER\Production

CleanMachinePackageLimitValue

Über diesen Eintrag wird die Anzahl installierter Windows Installer Pakete deklariert, ab welcher ein System nicht mehr zu einer *Clean Machine* gehören soll. Für den Import eines Paketes ist es wichtig, dass dieser Prozess auf einer *Clean Machine* erfolgt, da sonst die Wahrscheinlichkeit steigt, dass der *ActionState* falsch ermittelt werden könnte. Natürlich ist bei Paketen mit abgefüllter *SelfReg*-Tabelle (nur hier wird eine Pseudoinstallation auf der *Clean Machine* ausgelöst) eine vorgängige manuelle Installationen allfälliger technischer Abhängigkeiten (bspl. InstallScript Engine) möglich.

```
[Global]
CleanMachinePackageLimitValue=6
```

Ist die Anzahl auf dem Client installierter Windows Installer Pakete grösser als der in der *Settings.INI* deklarierte Wert, erscheint beim Importieren von Softwarepaketen eine Meldung auf dem Monitor, welche auf diesen Zustand hinweist.



Werden Softwarepakete auf einem mit vielen Anwendungen belegten Computer importiert, steigt die Wahrscheinlichkeit, dass die Ermittlung einiger Ressourcen einen falschen *ActionState* ergeben. Würden aus solchen Ressourcen Konflikte resultieren, dann würden diese durch eine automatische Korrektur nicht angepasst. **Sollten Sie die Pakete trotzdem generell auf einem mit Anwendungen belegten System importieren wollen, so wird das Markieren der Option *Use component request state instead of action state* (unter Optionen) empfohlen** (siehe [5.12.1 Optionen/Register General](#))! Dies verhindert auch das Anzeigen der obigen Fehlermeldung.

IsolationBeginRegistry

Mit diesem Eintrag kann global vorgegeben werden, ab wann eine Komponentenisolierung bei Registrykonflikten durchgeführt werden soll (default 15). (Kann durch die Benutzereinstellung (Options) übersteuert werden. Gilt daher nur als Initialisierungswert, wenn nicht die Standardeinstellung verwendet werden soll.)

```
[Global]
IsolationBeginRegistry=10
```

IsolationBeginFiles

Mit diesem Eintrag kann global vorgegeben werden, ab wann eine Komponentenisolierung bei Dateikonflikten durchgeführt werden soll (default 3). Siehe auch [5.12 Optionen in der Benutzeroberfläche des Conflict Explorers](#). (Kann durch die Benutzereinstellung (Options) übersteuert werden. Gilt daher nur als Initialisierungswert.)

```
[Global]
IsolationBeginFiles=2
```

Ausschnitt aus Options:

Resources Isolation - Start to Isolate

sum of resources minus registry conflicts to adjust > than:	15	(default 15)
sum of resources minus file conflicts to adjust > than:	3	(default 3)

TransformsAll

Entspricht der unten dargestellten Option (siehe auch [5.12 Optionen in der Benutzeroberfläche des Conflict Explorers](#)). Kann auf True oder False stehen und kann durch die Benutzereinstellung (Options) übersteuert werden. Gilt daher nur als Initialisierungswert.

```
[Global]
TransformsDefault=True
```

Settings for Manual Conflict Resolving

Apply all imported transform files (set as default)

Apply manually-selected transform files

TransformsAllReimport

Entspricht der unten dargestellten Option (siehe *Options*). Kann auf True oder False stehen. Dies bewirkt, dass beim Klick auf *Reimport* automatisch alle im Paketverzeichnis vorgefundenen Transformationen angewendet werden. (Kann durch die Benutzereinstellung (*Options*) übersteuert werden. Gilt daher nur als Initialisierungswert.)

```
[Global]
TransformsAllReimport=True
```

Re-Import and Automatic Resolve Settings

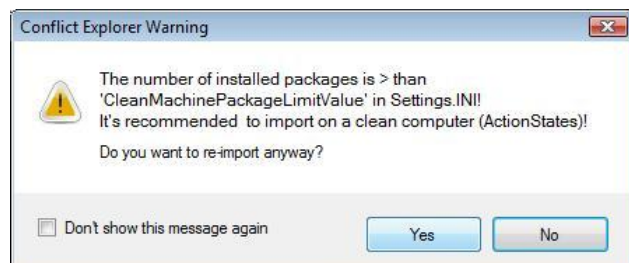
Apply TRANSFORMS=ALL as additional option (not set as default)

ActionStateToRequestState

Diese Einstellung ermöglicht, dass beim Importierprozess nicht der *Aktionsstatus* der Komponenten ermittelt wird, sondern der *Anforderungsstatus*. Ist der Eintrag auf True, wird geprüft, ob eine Komponente **zur Installation vorgesehen ist**. Steht der Eintrag hingegen auf False oder fehlt er komplett, dann wird geprüft, ob die Komponente durch Windows **installiert würde** (Standardeinstellung). Unterschiede ergeben sich vor allem, wenn die Komponente konditioniert ist

und die *Condition* False ergibt, die Komponente in einer höheren Version (*KeyPath*) bereits installiert ist oder die Komponente mit dem Attribut [msidbComponentAttributesNeverOverwrite](#) versehen ist.

Die Einstellung kann sinnvoll sein, wenn man den Importprozess auf Computer ausübt, welche bereits mit diversen Anwendungsinstallationen vorbelastet sind - wenn der Importprozess also nicht auf einer *Clean Machine* ausgeführt wird. Wird diese Einstellung verwendet, erscheint auch keine Meldung der nachfolgenden Art mehr beim Importieren der Softwarepakete.



Achtung: Die Ermittlung des *Aktionsstatus* (Standardeinstellung oder *ActionStateToRequestState=False*) ist auf jeden Fall die sicherere Methode, wenn es darum geht, zu ermitteln, ob eine Komponente in einer Umgebung effektiv installiert wird oder nicht. Daher sollten Sie die Einstellung *ActionStateToRequestState* nur in Ausnahmefällen auf *True* setzen. Auf der anderen Seite erkaufte man sich mit der Einstellung auf *True* im schlechtesten Fall, dass der *Conflict Explorer* bei einer automatischen Korrektur gewisse Konflikte zuviel korrigiert, deren Behebung also eigentlich gar nicht nötig wäre. Das dem *Conflict Explorer* zugrundeliegende Prinzip „Korrigiere soviel wie nötig, aber so wenig wie möglich!“ wäre hier also nicht 100%ig umgesetzt.

Die Einstellung kann auch über die Optionen temporär von jedem Benutzer eingestellt werden, sofern die Einstellung in der *Settings.INI* fehlt. Soll nicht generell der *RequestState*, anstatt der *ActionState* ermittelt werden, dann ist dies die empfohlene Implementation! Siehe auch [5.12 Optionen in der Benutzeroberfläche des Conflict Explorers](#). **Steht der Eintrag auf True, wird dieser durch die Benutzereinstellung (Options) im Gegensatz zu den vorangegangenen Einträgen nicht übersteuert! Achtung: Die Einstellung hat keinen Einfluss auf bereits importierte Softwarepakete!**

```
[Global]
ActionStateToRequestState=True
```

```
Import Settings
 Use component request state instead action state (not set as default)
```

AllowedToDelete

Hier werden Benutzer aufgeführt, die berechtigt sind, Pakete zu löschen, verschieben und *Gruppen* zu erstellen. Die Liste kann einfach erweitert werden. Grundsätzlich wird durch das Setup nur der Domänenbenutzer eingetragen, der das Setup ausgeführt hat. Erweiterungen müssten durch den Administrator durchgeführt werden.

```
[Security]
AllowedToDelete=oberlind;username;username
```

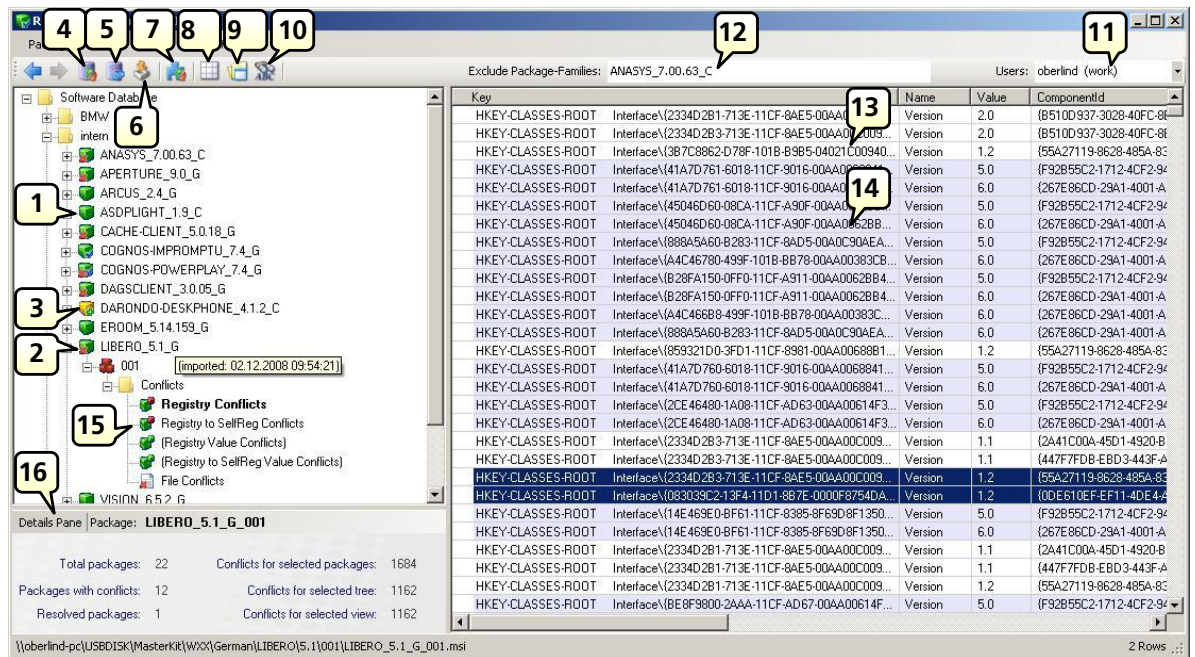
Hinweis:

Die *Settings.INI* lässt sich nicht über die *Conflict Explorer* Benutzeroberfläche abfüllen und sollte nur von einem Administrator durchgeführt werden. Um fehlerhafte Settings zu verhindern, werden entsprechende ACL-Berechtigungseinschränkungen auf diese Datei oder den ganzen *Conflict Explorer* Ordner empfohlen.

5. Bedienung des Conflict Explorers

In diesem Kapitel wird hauptsächlich auf Funktionen und das Handling innerhalb der Benutzeroberfläche eingegangen.

5.1 Die Benutzeroberfläche



Legende :

- | | |
|---|--|
| ① Paket ohne Konflikte | ⑨ Öffnen des Dateipfades mit Explorer |
| ② Paket mit Konflikten | ⑩ Suche nach Konfliktinformationen |
| ③ Neu importiertes oder reimportiertes Paket | ⑪ Userliste, wer mit dem <i>Conflict Explorer</i> arbeitet |
| ④ Ermittelt Konflikte (one to all) Dies sollte das Standardverfahren sein, wenn Konflikte von einem einzelnen Paket ermittelt werden. | ⑫ <i>Incompatibilities-Liste</i> : Mit all diesen Paketen erfolgt keine Konfliktprüfung (<i>Packagefamily-Exclude</i>) |
| ⑤ Ermittelt Konflikte (all to all) | ⑬ Konfliktanzeige |
| ⑥ Reimportiert das Software-Paket | ⑭ Konfliktanzeige (blau: <i>ActionState=False</i>) |
| ⑦ Automatische Konfliktanpassung | ⑮ Verschiedene Konfliktknoten, gruppiert nach Typ |
| ⑧ Öffnen der MSI mit Orca oder anderem MSI-Tool | ⑯ Detail Pane: Diverse Informationen |

5.2 Importieren von Softwarepaketen

Bevor wir Konflikte zwischen Softwarepaketen darstellen oder korrigieren können, müssen diese zuerst in die *Conflict Explorer* Datenbank importiert werden. Zu diesem Zweck benötigt *der Conflict Explorer* Informationen, wo sich die MSI-Datei befindet, ob und welche Transformationen beim Importierprozess angewendet werden müssen, wie das Softwarepaket in der *Conflict Explorer* Datenbank zu bezeichnet ist, welcher *Familyname* dieses Softwarepaket verwenden wird und optional, welcher *Gruppe* das Softwarepaket zugewiesen werden soll.

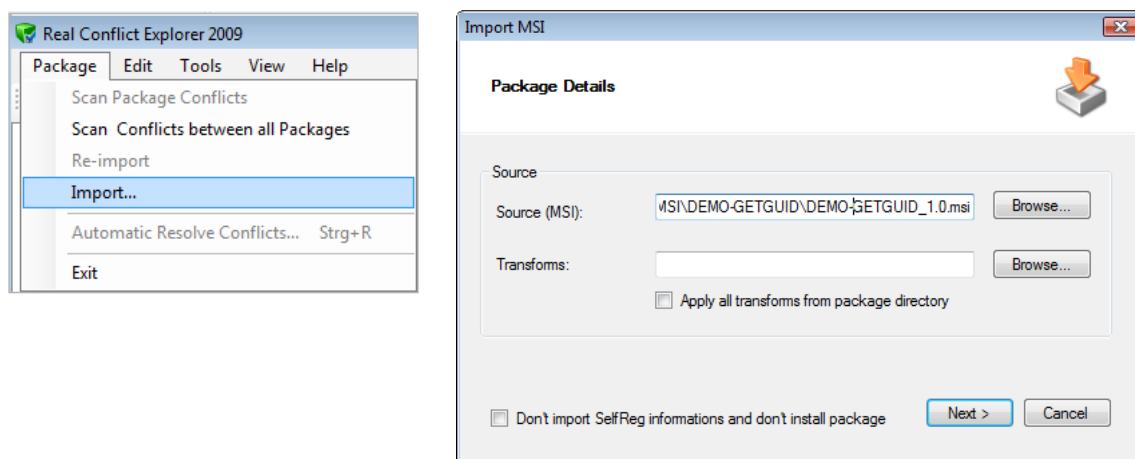
Wir unterscheiden 3 verschiedene Arten des Imports

1. Importieren eines Softwarepakets über die Benutzeroberfläche des *Conflict Explorers*
2. Importieren eines Softwarepakets mittels *ImportMsi.EXE* über Kommandozeilenoptionen (siehe Kapitel [4.2.1 ImportMsi.EXE](#) und Kapitel [6 Kommandozeilenoptionen](#)). Dieser Aufruf kann auch in ein Wise-Projekt eingebunden sein.
3. Mehrfaches Importieren von Softwarepaketen über ein Script wie *MultiPackageImport.vbs*

5.2.1 Importieren eines Softwarepakets über die Benutzeroberfläche

Dies ist die einfachste Variante, ein einzelnes Softwarepaket zu importieren. Dennoch ist für den generellen Ablauf in der Softwarebereitstellung eine automatisierte Variante vorzuziehen (siehe Punkt 2 in grauem Kasten des vorherigen Absatzes).

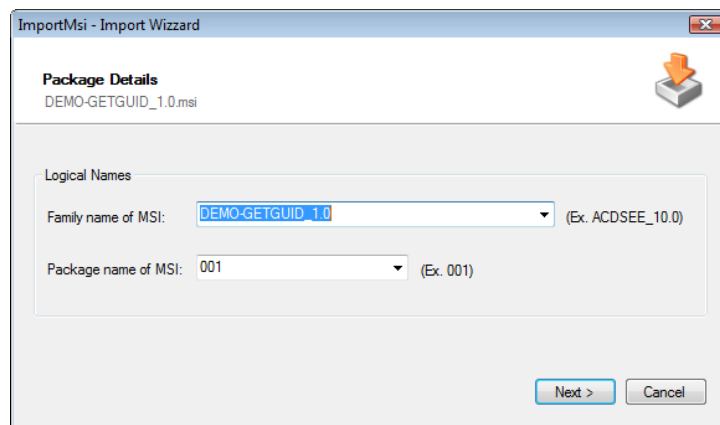
Zum Importieren über die Benutzeroberfläche des *Conflict Explorers* wählen Sie *Package/Import...*



Im darauf erscheinenden Dialogfeld wird der Name der MSI Datei abgefragt. Mittels der Schaltfläche *Browse...* können Sie zum Ablageort des Softwarepaketes navigieren. Eine Transformation (zweite Zeile) ist nur dann anzugeben, wenn für dieses Produkt eine MST-Datei existiert, welche für den Softwareverteilungsprozess vorgesehen ist. Alternativ dazu bietet sich an, in Umgebungen, wo die Transformation im Ablageverzeichnis des Softwarepakets gespeichert ist, die Option *Apply all transforms from package directory* anzuwählen. Wird diese Option gewählt, werden alle im Paketverzeichnis befindlichen MST-Dateien beim Importprozess in alphabetischer Reihenfolge angewendet, unabhängig davon, ob, und wenn ja, wieviele Transformationsdateien dort gespeichert sind.

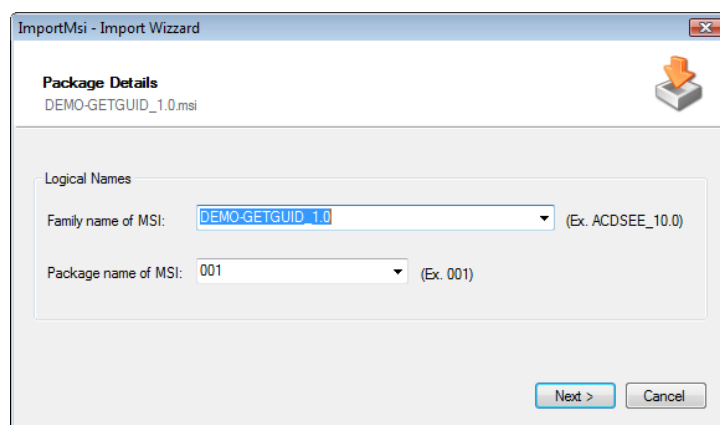
Die Option *Don't import SelfReg informations and don't install package* sollten Sie nicht markieren. Die Verwendung dieser Option ist dafür vorgesehen, in Ausnahmefällen den Importprozess für *SelfReg*-Informationen, und die damit verbundene Pseudoinstallation zu unterbinden. Denkbar ist ein solches Szenario, wenn die Pseudoinstallation beispielsweise einen unaufgeforderten Neustart des Systems veranlasst oder anderweitig den Importprozess stört. Die Option wird nur dann sichtbar, wenn der Verbund der MSI-Datei, inkl. MST-Datei/en *SelfReg*-Informationen enthält.

Im weiteren Verlauf werden die Namensbezeichnungen abgefragt. Sollten diese über die Schlüsselwörter *PropertyPackage* und *PropertyFamily* in der *Settings.INI* deklariert sein und das Installationspaket würde diese Properties in der *Property*-Tabelle enthalten, so wäre das Dialogfeld bereits vollständig abgefüllt. Fehlen ihm die Namen, so sind diese nun einzutragen. Auskunft über die Gestaltung der *Family* und eines *Package* gibt Kapitel [3 Terminologie](#). Beachten Sie auch, dass ein einmal importiertes Softwarepaket nicht mehr umbenannt werden kann (siehe dazu [5.19 Häufige Fragen](#), 3. Beitrag)



5.2.2 Automatischer Import mehrerer Softwarepakete in bestehenden Umgebungen

Im Verzeichnis `<Conflict Explorer>\Client Setup` befindet sich eine vb-Script-Datei (*MultiPakageImport.vbs*), die es in Unternehmen ermöglicht, aus einem Ablageverzeichnis bestehende Softwarepakete ohne Benutzerinteraktion zu importieren. Dieses Script ist als **Vorlage für eigene Ausfertigungen** gedacht. Denn der Aufbau der Ablagestruktur, der Speicherort der zugehörigen Transformationsdateien und die Regeln zur Namensbildung unterscheiden sich zum Teil deutlich in verschiedenen Unternehmungen. Das Vorlagenscript geht zur Namensbildung von abgefüllten *Properties* in der *Property*-Tabelle aus. Denkbar wäre auch, dass man die Namen der *Packagefamily* und des *Package* durch anderweitigen Mechanismen ermittelt und diese dann dem Aufruf von *ImportMsi.EXE* mittels der Kommandozeilenparameter `FAMILY=<XY> PACKAGE=<001>` übergibt. Für alle Softwarepakete, wo *ImportMsi* keine gültigen Namen übergeben werden und wo keine *Property*implementationen in der MSI-Datei vorliegen, erscheint der Benutzerdialog mit der Aufforderung, die Namen bekanntzugeben.



Real Packaging GmbH unterstützt Sie bei der Erstellung eines für Ihre Umgebung gültigen Import-Scripts. Informieren Sie sich kostenlos und unverbindlich über support@realpackaging.net.

5.3 Grundsätzlicher Standard-Ablauf

1. Nach Fertigstellung eines Softwarepakets importiert der Software-Paketierer das Software-Paket über ein *Wise-Projekt* durch Klick auf den entsprechenden Projekteintrag oder über *Package/Import...* aus der *Conflict Explorer* Benutzeroberfläche.

Hinweis:

Das Importieren muss auf diesem Betriebssystem erfolgen, worauf das Softwarepaket zielt! Vista-Pakete müssen demnach unbedingt auf einer **Clean Machine** von Windows Vista importiert werden!

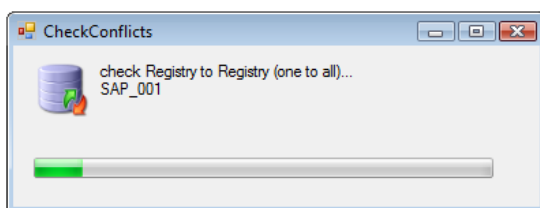
2. Im linken Fensterteil der *Conflict Explorer* Benutzeroberfläche ist in der explorerähnlichen Struktur das Paket auszuwählen (durch das Eingeben des ersten Buchstabens des Softwarepakets wird dieses sogleich selektiert), worauf man anschliessend auf den *Conflicts* Knoten navigiert.

Optisch werden importierte, aber noch nicht gescannte Pakete durch ein gelbes *Packagefamily*-Icon dargestellt.



Durch die Navigation wechselt der Status des Icons *Scan Package Conflicts* in der Symbolleiste.

3. Daraufhin ist auf das Icon *Scan Package Conflicts* in der Symbolleiste zu klicken. Dadurch werden die Ressourcen dieses Pakets mit den Ressourcen aller importierter Pakete verglichen. Welche Standardregeln während diesem Prozess angewendet werden, finden sich unter [5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten](#).



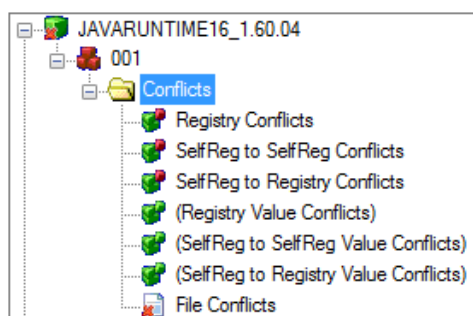
4. Konflikte können nun analysiert und organisatorische Massnahmen (siehe [5.5 Ausschluss von Konflikten](#)) getroffen und/oder eine Konfliktkorrektur vorgenommen werden. (siehe [5.6 Automatische Konfliktkorrektur](#) und [5.7 Manuelle Konfliktkorrektur](#))

5.4 Konfliktarten

Der *Conflict Explorer* bringt verschiedene Arten von Konflikten zur Anzeige. Diese gruppieren sich in den Ressourcentyp (*Registry*, *SelfReg*, *File*), sowie aus Kombinationen daraus (bspl. *Registry to SelfReg*) und unterscheiden sich weiter in der Eigenschaft, ob es sich um Konflikte handelt, die aufgrund verschiedenen Inhaltstypen (*content*) der gleichen Ressource (*Value Conflicts*) zum Problem führen oder um Konflikte, die bei einer Deinstallation eines Softwareproduktes ein unbeabsichtigtes Fehlverhalten verursachen.

Nicht geprüft und dargestellt werden Pseudokonflikte aufgrund der Verwendung verschiedener Dateiversionen der gleichen Datei in unterschiedlichen Softwarepaketen, wie dies aus anderen Authoringtools bekannt ist. Werden minimale Richtlinien im Design und im Einsatz solcher Dateien bei der Applikationsentwicklung eingehalten, so darf ein Upgrade einer gleichen Datei nie zu einem Problem ausarten. Windows Installer garantiert durch die [File Versioning Rules](#), dass jeweils die aktuellste Version auf dem System gewinnt.

Hervorzuheben sind insbesondere die *SelfReg*-Konflikte, die bisher von keiner anderen Konkurrenz-Software erkannt werden. Dennoch führen gerade solche Konflikte oft zu einem Problem, dass nach der Deinstallation eines Softwareproduktes ein ungültiger Installationsstatus einer anderen Anwendung vorliegt, bzw. das Entfernen von gemeinsamen Ressourcen zum nicht mehr Funktionieren anderer Applikationen führen kann.



5.4.1 Registry Conflicts:

Dies sind Konflikte, wo der Registrykey (*Root, Key, Name*) zwischen dem aktuell gewählten Paket gegenüber anderen Paketen gleich ist und die unter [5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten](#) vermerkten Bedingungen zutreffen. Man beachte die hier etwas inkonsequente Namensgebung. Eigentlich müsste es „*Registry to Registry Conflicts*“ heißen. Der schnelleren Erkennung wegen (kürzerer Text) und weil es sich hier um den Knoten mit den meisten Konflikten handelt, wurde er aber auf *Registry Conflicts* belassen.

5.4.2 SelfReg to SelfReg Conflicts

Hierbei handelt es sich um Konflikte, die aus *SelfReg*-Registrierungen gegenüber anderen *SelfReg*-Registrierungen entstehen. Best-Practice Regeln sprechen sich ganz klar gegen den Einsatz der *SelfReg* Tabelle in MSI-Dateien aus. Stattdessen sollte man die Entsprechungen in der *Registry*-Tabelle plazieren (siehe [Rule 19](#)). Trotzdem finden wir immer wieder abgefüllte *SelfReg*-Tabellen, die zu einem Problem werden können. Diese Probleme zu verhindern, dies ist ein Ziel des *Conflict Explorers*.

5.4.3 SelfReg to Registry Conflicts

Gleich wie oben, nur dass *SelfReg*-Informationen gegen *Registry* Informationen verglichen werden.

5.4.4 Registry to SelfReg Conflicts

Gleich wie oben, nur dass *Registry* Informationen gegen *SelfReg*-Informationen verglichen werden.

5.4.5 (Value Conflicts)

Für alle Registrykonfliktarten gibt es auch entsprechende *Value Conflicts*. Diese sind

1. (*Registry Value Conflicts*),
2. (*SelfReg to SelfReg Value Conflicts*),
3. (*SelfReg to Registry Value Conflicts*) und
4. (*Registry to SelfReg Value Conflicts*).

Wie bereits besprochen, handelt es sich hier um Konflikte, wo der selbe *Key* in unterschiedlichen Paketen Verwendung findet, aber unterschiedliche Inhalte aufweist. Durch eine automatische Konfliktkorrektur werden diese Konflikte nicht bereinigt und verbleiben in der Ansicht. Hier besteht auch kein zwingender Korrekturbedarf. Jeder Fall muss separat analysiert werden.

Hinweis:

Zu beachten ist insbesondere, dass Konflikte jeweils in verschiedenen Konfliktknoten erscheinen können. D.h. dass beispielsweise ein als *Value Conflict* ausgewiesener Konflikt auch als *Registry Conflict* angezeigt werden kann (das eine hat nichts mit dem anderen zu tun!)

Hinweis:

Werden gleichartige Dateien mit *SelfReg*-Code in verschiedenen Softwarepaketen in unterschiedlichen Verzeichnissen geführt und verweisen *SelfReg*-Keys auf den Dateipfad der COM-Datei, so finden wir diese unterschiedlichen Pfade in den *Value Conflicts* wieder. In solchen Fällen wird empfohlen, solche Dateien zentral abzulegen. Beispielsweise unter %SystemRoot%\System32, eine entsprechende *Incompatibility* im *Conflict Explorer* zu setzen (sofern eine gemeinsame Installation dieser Softwarepakete ausgeschlossen wird) oder eine *.LOCAL Isolation* vorzunehmen.

5.4.6 File Conflicts

Dies sind Konflikte zwischen Dateien, d.h. wo eine gleich benannte Datei aus dem selben Verzeichnis in mehreren Paketen vorkommt und die unter [5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten](#) vermerkten Bedingungen zutreffen.

5.5 Ausschluss von Konflikten

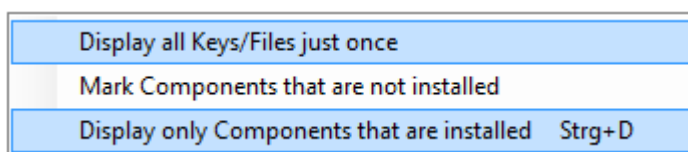
Es gibt mehrere Möglichkeiten, Konflikte auszufiltern oder eine Anzeige vorweg auszuschliessen. Diese Möglichkeiten bestehen in anderen Konflikt Management Lösungen nicht.

Zunächst ist zwischen Filteroperationen und organisatorischen Massnahmen für den Ausschluss von Konflikten zu unterscheiden.

5.5.1 Filteroperationen

Filteroperationen sind temporärer Natur und ermöglichen das leichtere Lesen der Konfliktanzeige. Diese dienen dazu, das Handling zu vereinfachen und/oder die Transparenz zu erhöhen.

Folgende Kontextmenüeinträge ermöglichen solche Filteroperationen (im Bild blau markiert). Diese Menüeinträge finden wir auch unter dem Menü *View*.



Der erste Eintrag ermöglicht die Darstellung der betroffenen Konflikte in dieser Form, dass der Konflikt nur einmal im Anzeigefenster erscheint. Denn oft kann es vorkommen, dass der selbe Konflikt mehrfach aufgeführt ist, weil er zu mehreren Paketen konfliktierend ist.

Der untere Menüeintrag verhindert, dass Konflikte zur Anzeige gelangen, wenn die entsprechenden Ressourcen durch das Produkt nicht installiert würden (*ActionState=False* oder *ActionState* des konfliktierenden Pakets auf False). Standardmässig ist ja nur eine optische Differenzierung durch eine blaue Hintergrundfarbe der Zellen gewählt.

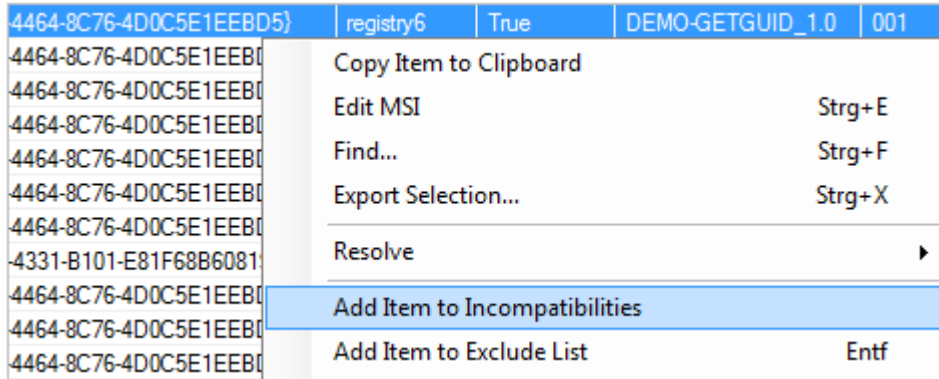
5.5.2 Organisatorische Massnahmen

Organisatorische Massnahmen definieren bleibende Einstellungen (bis zur nächsten Anpassung), um Konflikte auszuschliessen **ohne das Produkt zu verändern**. Diese Massnahmen haben also keinen Einfluss auf das Verhalten des Paketes bei der Installation oder Deinstallation, sondern ermöglichen es die Standardregeln für den Ausschluss von Konflikten um die Prüfung gegenüber bestimmten

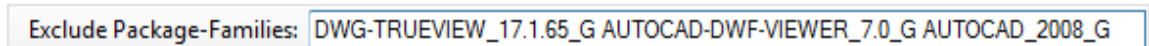
Softwarepaketen zu erweitern. Siehe auch ([5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten](#))

Packagefamily-Exclude

Über den Kontextmenüeintrag, das beim Klick mit der rechten Maustaste auf einen Konflikt erscheint, kann mit *Add Item to Incompatibilities*, die *Packagefamily-Excludeliste* mit dem *Familynamen* erweitert werden. (Der Mauszeiger muss nicht zwingend auf dem Paketnamen stehen.)



Nach der Operation findet man den *Familynamen* in der Symbolleiste angezeigt und die Konfliktanzeige reduziert sich um alle Konflikte, die mit der ausgeschlossenen Familie ausgewiesen waren.

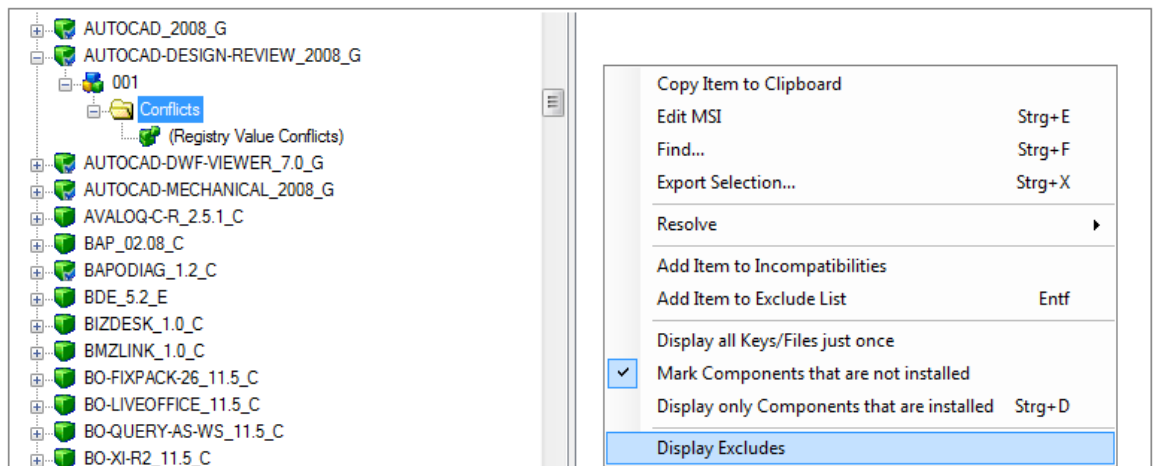


Zum Löschen eines Eintrages kann man direkt in der Textbox Korrekturen vornehmen und diese mit Enter abschliessen. Auch hier wird wieder ein *Rescan* durchgeführt, um die Konfliktanzeige zu aktualisieren.

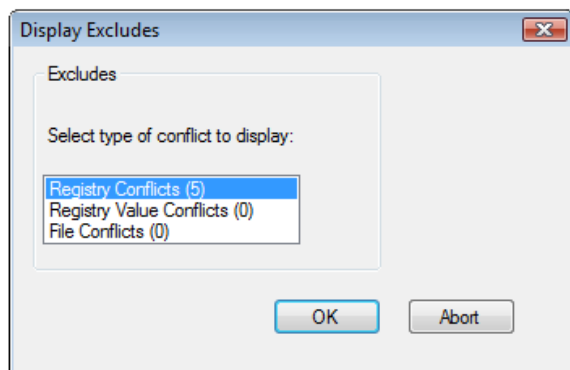
Excludelist

Mittels dem Hinzufügen ausgewählter Konflikte zu der *Excludelist* (siehe auch letztes Bild des Kontextmenüs – *Add Item to Excludelist*) kann man selektiv einzelne Konflikte aus der Anzeige ausschliessen. Diese Operation ist auch durch eine Auswahl und der Taste durchführbar.

Sollten aufgrund kompletter Ausschlüsse keine Konfliktknoten mehr in der Exploreransicht erscheinen, dann kann über die Auswahl des Knotens *Conflicts* das Kontextmenü in der Konfliktanzeige (rechtes Fensterelement) aufgerufen werden, um dort *Display Excludes* zu wählen.



Daraufhin erscheint folgendes Fenster, wo die Art der ausgeschlossenen Konflikte ausgewählt werden kann.



5.5.3 Empfohlene Anwendung

- Grundsätzlich sollte man die Softwarepakete in *Gruppen* ordnen, die jede ein Set von Computertypen repräsentiert. Konflikte werden nur dann dargestellt, wenn Konflikte zu Paketen in der selben *Gruppe* existieren. Wird das selbe Softwarepaket in einer anderen *Gruppe* mit der selben Namensbezeichnung geführt, so werden zusätzlich die Konflikte dieser anderen *Gruppe* bei der Konfliktdarstellung miteinbezogen!
- Sollte bei einem Paket durch anderweitige Steuerungsinstrumente gewährleistet sein, dass dieses Softwarepaket nicht gemeinsam mit einem anderen auf ein System installiert wird, dann sind diese Regeln auch im *Conflict Explorer* über die *Incompatibilities* abzubilden. Dies wird durch eine Erweiterung der *Packagefamily-Excludeliste* beim entsprechenden Paket bewerkstelligt.

Hinweis:

Man beachte, dass bei jedem neuen *Package* die *Packagefamily-Excludeliste* neu erstellt werden muss. Die Gültigkeit erstreckt sich also nicht auf die ganze *Packagefamily*!

- Findet man andere Konstellationen, wo man sicher ist, dass das im *Conflict Explorer* ausgewählte Paket nie mit dem konfliktierenden Paket installiert wird, dann ist ebenfalls die *Packagefamily-Excludeliste* zu erweitern.
- Auch in Fällen, wo das im *Conflict Explorer* ausgewählte Paket nur gemeinsam mit dem konfliktierenden Paket installiert **und deinstalliert** wird, kann man die *Packagefamily-Excludeliste* erweitern, sofern man an allfälligen Informationen der *Value-Conflicts* nicht interessiert ist.
- Für selektive Konflikte, die man aus anderen Gründen aus der Ansicht ausschliessen will, ist die Funktion *Add Item to Excludelist* empfohlen (Kontextmenü auf Konflikteintrag). Hier kann man in einem erweiterten Dialog den Grund weiter beschreiben, so dass dieser bei repetitiven Analysen auch anderen Software-Paketierern plausibel wird.

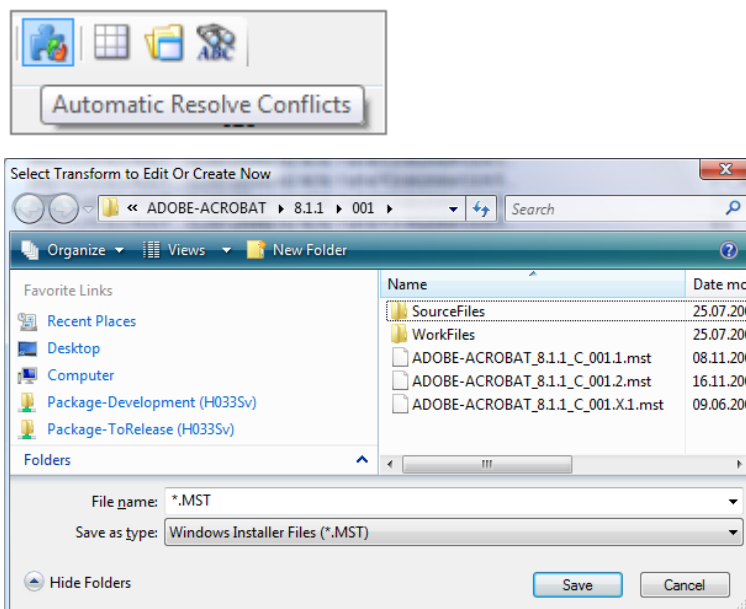
5.6 Automatische Konfliktkorrektur

Die automatische Konfliktkorrektur sollte das Standardverfahren sein, um Konflikte zu bereinigen. Während diesem Prozess werden alle Konflikte, ausser die *Value Conflicts* Konflikte korrigiert und die Korrekturen in eine Transformation (MST-Datei) geschrieben.

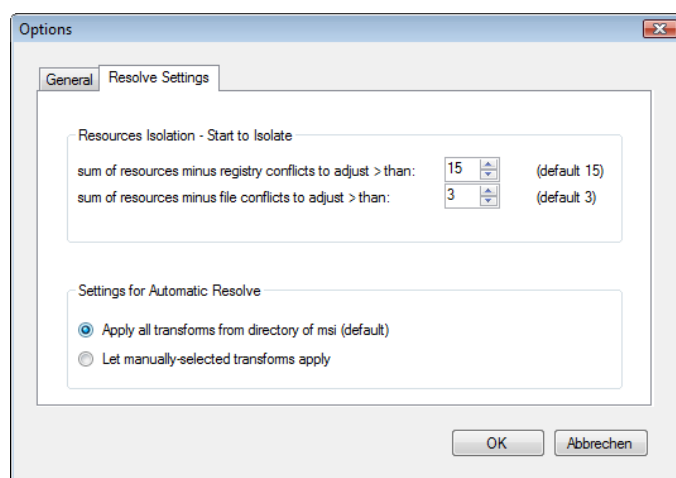
Hinweis:

Vor einer Konflikthanpassung ist darauf zu achten, dass ein *Scan Package Conflicts* kurz zuvor durchgeführt wurde. Denn die Anzeige der Konflikte könnte nicht mehr aktuell sein, wenn seit dem letzten *Scan Package Conflicts* Gruppen-Verschiebeoperationen getätigt wurden, neue Pakete importiert oder bestehende reimportiert, sowie existierende aus der Datenbank entfernt wurden!

Nach dem Klick auf *Automatic Resolve Conflicts* wird ein Dialog angezeigt, welcher nach der Bezeichnung der zu erstellenden Transformation fragt.



Dabei werden Einstellungen aus dem Menü *Tools/Options* angewendet. Die Standard-Einstellungen sind unten abgebildet.



Hinweis:

Wird im Dialog *Select Transform to Edit or Create Now* eine bestehende MST-Datei ausgewählt, so werden die bestehenden Inhalte aus dieser MST-Datei behalten. Dadurch können also auch MST-Dateien mit neuen Konflikthanpassungen erweitert werden!

Bei der automatischen Konfliktkorrektur wird zunächst versucht, einen Abgleich der *ComponentId* zu bewerkstelligen. Wenn dies misslingt, wird eine Konflikthanpassung mit Standardeinstellungen (bei Dateien *SharedDll Attribute* setzen, ansonsten Setzen des *Permanent Attributes*) angewendet. Entsprechend der verwendeten Einstellungen wird bei Vorfinden eines Komponentendesigns, wo nicht alle Ressourcen der Komponenten konfliktauslösend sind, gleichzeitig eine *Komponentenisolation* der konfliktierenden Ressourcen durchgeführt.

Ebenfalls werden gewisse Voreinstellungen verwendet. So vollzieht der *Conflict Explorer* nur Anpassungen an Komponenten, die nicht blau eingefärbt sind, d.h. die bei beiden Paketen (aktuelles Paket und konfliktierendes Paket) zur Installation vorgesehen sind, sprich der *ActionState* auf True steht.

Hinweis:

In bestimmten Ausnahmefällen (beispielsweise, wenn in einer Komponente verschiedene Ressourcentypen konfliktierend sind) kann es vorkommen, dass im ersten Durchlauf noch Restkonflikte in der Darstellung übrig bleiben. In solchen Fällen ist einfach die automatische Konfliktkorrektur noch einmal anzuklicken und im Dialogfeld die soeben erstellte MST-Datei auswählen.

5.6.1 Besonderheit bei Registrierungsverweisen auf Dateien der Installationsinstanz

Neben den oben beschriebenen Regeln, verwendet der *Conflict Explorer* in gewissen Situationen zusätzliche Regeln. Diese Modifikationen unternimmt der *Conflict Explorer* sowohl bei einer automatischen, als auch bei einer manuellen Konfliktkorrektur:

Für alle Arten von *Registry*- und *SelfReg*-Konflikten (keine *Value Conflicts*), die auf...

1. ...Dateien verweisen, die mit der selben MSI installiert werden und...
2. ...wo die verwiesenen Dateien nicht schon zu einer angepassten Komponente gehören,

wird für die Dateiressourcen eine Konflikthanpassung mit dem *Permanent*-Attribut (*msidbComponentAttributesPermanent*) vorgenommen. Diese zusätzliche Manipulation erfolgt auch dann, wenn die Dateien selbst nicht im *File Conflicts*-Knoten erscheinen, da sie beispielsweise gegenüber dem konfliktierenden Paket in einem anderen Verzeichnis gespeichert sind.

Durch diesen zusätzlichen Schutz soll gewährleistet werden, dass bei COM-Dateien mit Registryverweisen in der *SelfReg*- oder *Registry*-Tabelle neben den gemeinsamen Registrierungsschlüsseln auch die dazugehörigen Dateien auf dem System belassen werden, wenn es zu einer Deinstallation oder zum Upgrade eines einzelnen Softwarepakets kommt.

Im umgekehrten Fall wird bei Dateikonflikten auf die...

1. ...in Registrierungseinträgen aus *Registry*- oder *SelfReg*-Konflikten verwiesen wird und...
2. ...wo die verweisenden Registrierungsschlüssel nicht zu einer angepassten Komponente gehören,

keine zusätzliche Anpassung der Registrierungsressourcen vorgenommen. Der Grund hierfür liegt darin begründet, dass nicht programmtechnisch und automatisch bestimmt werden kann, welche Registrykeys neben den verweisenden Registrierungsschlüsselressourcen alle zusätzlich zu diesen COM-Komponenten dazugehören müssten. Zudem ist dieser Fall unwahrscheinlich, denn bei einem Dateikonflikt mit einer Datei, die auf sich bezogene Registrierungsschlüssel verwendet, würden zusätzlich *Registry*- oder *SelfReg*-Konflikte in den entsprechenden Konfliktknoten erscheinen (sofern die Komponente, worin die Registrierungsressourcen enthalten sind, nicht schon richtig markiert ist) und eine reguläre Konflikthanpassung wäre dadurch gewährleistet. Natürlich nur sofern die Dateien aus den unterschiedlichen Softwarepaketen die gleichen Registrierungsschlüssel verwenden würden, andernfalls wäre eine zusätzliche Anpassung sowieso hinfällig.

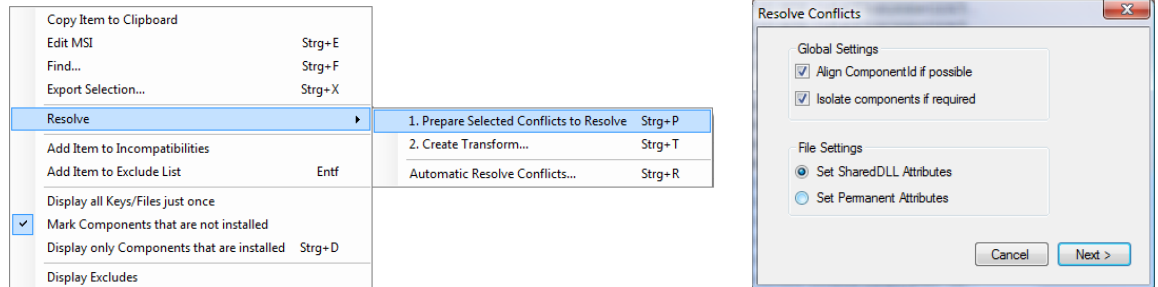
Achtung: Dieses Feature kann über die Optionen *Tools/Options...* ein-, bzw. abgeschaltet werden!

General Settings

Set 'Permanent' attribute to files if registry references exists (set as default)

5.7 Manuelle Konfliktkorrektur

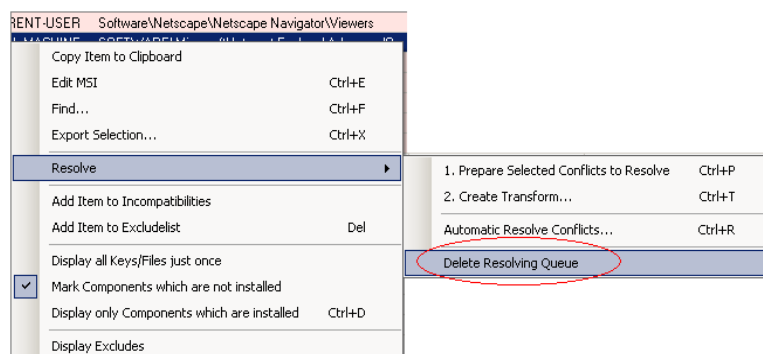
Über die manuelle Konfliktkorrektur lassen sich benutzerdefinierte Konflikthanpassungen vornehmen. So können die Einstellungen für jeden gewählten Konflikt geändert werden.



Zudem kann man die Auswahl der Konflikthanpassung erweitern oder beschränken. Hierzu werden die zu korrigierenden Konflikte markiert und im Kontextmenü der Befehl *1. Prepare Selected Conflicts to Resolve* ausgewählt.

Alle zur Konfliktbereinigung vorgesehenen Konflikte werden zur besseren Erkennlichkeit rot eingefärbt. Beachten Sie, dass die Wahl einer einzigen Zeile für die Konfliktkorrektur, das Einfärben mehrerer Zeilen in der Konflikthanzeige zur Folge haben kann. Dies rührt daher, dass ein einzelner Konflikt mehrfach zur Anzeige kommen kann, weil er gegenüber mehreren Softwarepaketen konfliktierend ist.

Über *2. Create Transform...* kann die Konfliktkorrektur schliesslich angewendet werden und der Benutzer wird zur Eingabe der Basis-MSI, der Transformationen, welche vor der Konfliktkorrektur angewendet werden sollen und zur Bezeichnung der zu erstellenden Transformation aufgefordert.



Delete Resolving Queue

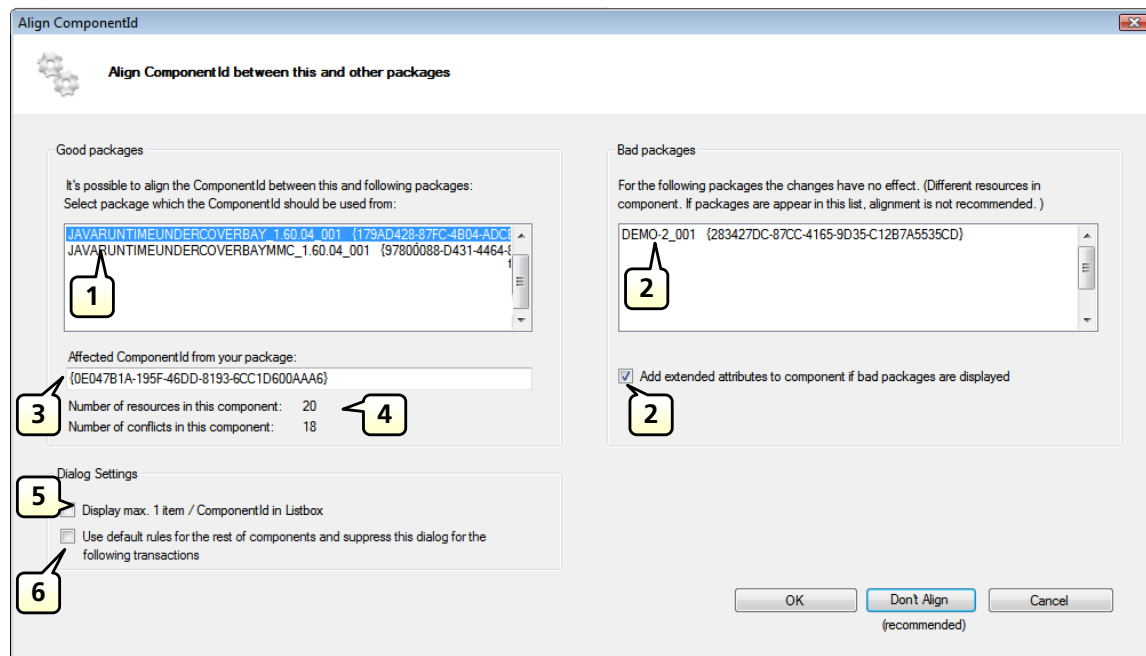
Klicken Sie auf das Kontextmenü *Delete Resolving Queue* über einem roten Eintrag, so kann die zur Korrektur vorgesehene Warteschlange gelöscht werden (siehe Bild oben). Achtung: Hierbei wird die komplette Warteschlange des entsprechenden Softwarepakets gelöscht! Nach der Operation erscheinen die Zeilen wieder in gewohntem schwarz auf weiss oder schwarz auf blau.

Achtung: Auch bei der manuellen Konfliktkorrektur werden die unter [5.6.1 Besonderheit bei Registrierungsverweisen auf Dateien der Installationsinstanz](#) beschriebenen Regeln angewendet.

5.8 Anpassungen der ComponentId

Beim Prozess der Angleichung der *ComponentId* wird geprüft, ob die jeweiligen konfliktauslösenden Komponenteninhalte identisch sind, so dass eine allfällige Anpassung nur dann erfolgt, wenn dies zutrifft. Findet der *Conflict Explorer* beim Anpassungsprozess genau ein Paket oder eine einzelne *ComponentId* aus mehreren Software-Paketen, die zu übernehmen ist, erfolgt die Anpassung automatisch ohne Benutzerinteraktion. Sollte sich die konfliktierende Ressource hingegen in mehreren Softwarepaketen finden, wo jeweils die konfliktierende Komponente den selben Aufbau aufweist, aber in den Softwarepaketen unterschiedliche *ComponentId*'s verwendet werden, wird der

Benutzer gefragt, zu welcher *ComponentId* eine Anpassung durchgeführt werden soll. Es erscheint dann ein Dialogfeld mit verschiedenen Informationen, die der Benutzer zu bestätigen hat. Beachten Sie, dass der *Conflict Explorer* standardmässig die Schaltfläche *OK* oder *Don't Align* aktiviert, je nachdem, ob „*Bad packages*“ (siehe weiter unten) vorgefunden wurden oder nicht.



Legende :

① Good packages

In diesem Listenfeld aufgelistete Softwarepakete verwenden ebenfalls die konfliktierende Ressource und der Komponenteninhalt ist identisch mit dem des anzupassenden Softwarepakets. Neben dem Namen finden wir die *ComponentId* angezeigt. Der Benutzer wird aufgefordert, hier seine Wahl vorzunehmen (Klick auf das entsprechend gewünschte Softwarepaket mit der entsprechenden *ComponentId*). Beachten Sie, dass hier gleichartige *ComponentIds* aus verschiedenen Softwarepaketen zur Darstellung kommen können. Durch Markieren der Checkbox *Display max. 1 item / ComponentId in Listbox* (5), wird die Darstellung reduziert, was das Auswahlverfahren vereinfacht.

② Bad packages

Hier werden informativ Softwarepakete aufgelistet, die zwar die konfliktierende Ressource enthalten, deren Aufbau aber nicht identisch ist. Ein Angleichen mit einem im Listenfeld *Good packages* aufgeführten Softwarepaket verhindert daher nicht, dass nach dem Abschluss der Operation das Softwarepaket konfliktfrei erscheint. Sollte die ausgewählte *ComponentId* aus *Good packages* gar identisch sein mit einem unter *Bad packages* aufgeführten Softwarepaket, dann wird ein Angleichen der *ComponentId* erst recht nicht empfohlen.

Findet der *Conflict Explorer* konfliktierende Softwarepakete, die er unter diesem Listenfeld darstellt, so wird die Schaltfläche *Don't Align* aktiviert, um dem Benutzer seine Empfehlung mitzuteilen. Zusätzlich wird dies durch den Text *recommended* unter der Schaltfläche verdeutlicht.

Standardmässig ist auch die Checkbox *Add extended attributes to component if bad packages are displayed* aktiviert. Wenn der Benutzer dennoch eine Angleichung der *ComponentId* anstrebt, wird mit dieser Option zusätzlich das Attribut *SharedDLL* bei Dateikonflikten oder das *Permanent*-Attribut bei Registrykonflikten gesetzt.

③ **Affected ComponentId from your package**

Darstellung der *ComponentId* der konfliktauslösenden Ressource des anzupassenden Softwarepakets.

④ Informationen der konfliktauslösenden Komponente des anzupassenden Softwarepakets.

⑤ **max. 1 item / ComponentId in Listbox**

Durch die Wahl dieser Option zeigt das Dialogfeld alle Einträge in *Good packages* pro *ComponentId* nur einmal an. Man kann die Checkbox selektieren und wieder abwählen, um die Unterschiede in der Darstellung erkennen zu können. Das Auswahlverfahren der abzugleichenden *ComponentId* soll damit vereinfacht werden. (Siehe auch *Good packages*)

⑥ **Use default rules for the rest of components and suppress dialog for the following transactions**

Die Wahl dieser Option verhindert ein weiteres Erscheinen dieses Dialogfeldes innerhalb der kompletten Anpassungstransaktion. Stattdessen werden Standardeinstellungen zum Abgleich der *ComponentId* verwendet, wenn der Dialog anschliessend mit **OK** bestätigt wird:

1. Wird im weiteren Verlauf nur eine Ziel-*ComponentId* gefunden und kein *Bad package* ermittelt, wird zu der gefundenen *ComponentId* abgeglichen.
2. Werden mehrere Ziel-*ComponentIds* (*Good package*) gefunden und keine *Bad packages* ermittelt, wird zu der am häufigsten vorgefundenen *ComponentId* abgeglichen.
3. Werden mehrere Ziel-*ComponentIds* (*Good package*) gefunden, die identisch sind und keine *Bad packages* ermittelt, dann wird zu dieser vorgefundenen *ComponentId* abgeglichen.

Beachten Sie, dass für die aktuell fokussierte Komponente eine Auswahl aus dem Listenfeld der *Good packages* erfordert oder die Aktion mit *Don't Align* abgeschlossen werden muss.

Nach der Wahl der Option *Use default rules...* und Bestätigung des Dialogfeldes mit **Don't Align** wird für die aktuelle und alle weiteren Komponenten, wo mehrere Ziel-*ComponentIds* (*Good package*) gefunden werden, keine Anpassung der *ComponentId* vorgenommen. Wird hingegen im späteren Transaktionsverlauf eine **einzelne** Ziel-*ComponentId* für eine anzugleichende Komponente gefunden und zu dieser kein *Bad package* ermittelt, erfolgt dort natürlich nachwievorr ein Abgleich zu der gefundenen *ComponentId*.

Sollten Sie während der Anzeige des Dialogfeldes Informationen aus der Konfliktdarstellung des Softwarepakets benötigen, können Sie den im Hintergrund befindlichen *Conflict Explorer* aktivieren und in der Konfliktansicht entsprechend navigieren oder andere Funktionen ausführen. Beachten Sie, dass sie während dieser Phase keinen zweiten Anpassungsprozess starten können. Zudem sollten Sie vor Abschluss ihrer Tätigkeiten über das Dialogfeld, in der Konfliktansicht wieder auf das ursprünglich fokussierte Softwarepaket wechseln, sofern Sie im TreeView zwischenzeitlich ein anderes Paket ausgewählt haben sollten.

5.9 Angewendete Standard-Regeln beim Ermitteln von Konflikten

Folgende Regeln werden beim Klick auf *Scan Package Conflicts* angewendet. Die Konfliktanzeige auf der rechten Seite der Benutzeroberfläche ist das Resultat dieser Prüf-Regeln:

1. Konflikte zwischen zwei Produkten aus der selben *FamilyBase* (Name vor dem „_“ gleich) werden ausgeblendet. Bspl: Konflikte zwischen **MOZILLA-FIREFOX_2.00.9**, **MOZILLA-FIREFOX_2.00.11** und **MOZILLA-FIREFOX_2.00.12** werden nicht angezeigt.
2. Konflikte zwischen Packages (bspl. 001 zu 002) sind keine Konflikte.
3. Konflikte zwischen Ressourcen aus Paketen, wo die zugrundeliegenden Komponenten die selbe *ComponentId* verwenden, sind keine Konflikte (ausser bei *Value Conflicts*).
4. Konflikte zwischen Ressourcen aus Paketen, wo die zugrundeliegenden Komponenten das Attribut *msidbComponentAttributesPermanent* verwenden, sind keine Konflikte (ausser bei *Value Conflicts*).
5. Konflikte zwischen Ressourcen aus Paketen, wo die zugrundeliegenden Komponenten das Attribut *msidbComponentAttributesSharedDllRefCount* verwenden, sind keine Konflikte (ausser bei *Value Conflicts*).
6. Filekonflikte, wo die Datei aus %WINDIR%\System32 stammt, sind keine Konflikte (diese trägt der Windows Installer automatisch in die *SharedDLL*-Liste ein).
7. Konflikte aus Paketen aus unterschiedlichen *Gruppen* werden ausgeblendet, sofern das fokussierte Paket nicht in beiden Gruppen vorkommt.
8. Konflikte, die in die *Excludelist* eingetragen wurden (Funktion des *Conflict Explorers*) werden ausgeblendet.
9. Konflikte, wo die *Packagefamily* in die *Packagefamily-Excludeliste* eingetragen wurde (Funktion des *Conflict Explorers*) werden ausgeblendet.
10. Registrykonflikte mit dem Zeichen „+“ in der ‚Name‘ Spalte werden ausgeblendet (entspricht einer Leeraanweisung für Windows-Installer, um *Keys* zu markieren, welche bei der Deinstallation nicht gelöscht werden)

5.10 Regeln beim Bilden des KeyPath bei neuen Komponenten

Werden durch die Konfliktkorrektur *Komponentenisolationen* durchgeführt, dann werden bei einer neuen *ComponentIsolationX* Komponente in der Transformation folgende Regeln bei der Erstellung des *KeyPath*-Eintrags in der angegebenen Reihenfolge angewendet (sobald eine Regel zutrifft, wird diese Regel und keine weiteren Regeln angewendet):

1. Hatte die ursprüngliche Komponente aus welcher die Ressourcen isoliert wurden, **keinen KeyPath**, dann verwendet auch die neue Komponente **keinen KeyPath**.
2. Befindet sich die Ressource des *KeyPath* der alten Komponente nun in der isolierten Komponente, so wird dieser *KeyPath* in der neuen Komponente *ComponentIsolationX* verwendet.
3. Enthält die neue Komponente *File* Ressourcen, so gewinnt die erstermittelte EXE-Datei. Wird keine EXE-Datei gefunden, wird die erstermittelte DLL-Datei verwendet. Wird keine DLL-Datei vorgefunden, wird die erstermittelte Datei mit unbekannter Extension als *KeyPath* verwendet.
4. Konnte noch kein *KeyPath* ermittelt werden, so wird ein Registry-*KeyPath* verwendet, sofern der Eintrag nicht folgendem Schema entspricht:
 - a. Value=0 UND
 - b. (Name= + ODER Name= - ODER Name= *)

Siehe [msidbComponentAttributesRegistryKeyPath](#)

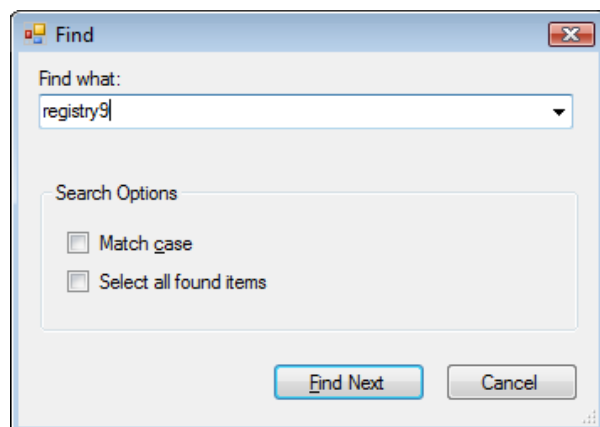
...If the Value field of the corresponding record in the Registry table is null, the Name field in that record must not contain "+", "-", or "*". For more information, see the description of the Name field in Registry table.

Folgende Regeln gelten für die alte Komponente, woraus die Ressource/n entfernt wurden:

1. Hatte die ursprüngliche Komponente aus welcher die Ressourcen isoliert wurden **keinen KeyPath**, dann ergibt sich keine weitere Operation.
2. Befindet sich die Ressource des *KeyPathes* der alten Komponente nach der Isolationstransaktion immer noch in der alten Komponente, dann ergibt sich keine weitere Operation.
3. Enthält die alte Komponente *File* Ressourcen, so gewinnt die erstermittelte EXE-Datei. Wird keine EXE-Datei gefunden, wird die erstermittelte DLL-Datei verwendet. Wird keine DLL-Datei vorgefunden, wird die erstermittelte Datei mit unbekannter Extension als *KeyPath* verwendet.
 - c. Konnte noch kein KeyPath ermittelt werden, so wird ein Registry-Keypath verwendet, sofern der Eintrag nicht folgendem Schema entspricht:
Value=0 UND
 - d. (Name= + ODER Name= - ODER Name= *)

5.11 Suche

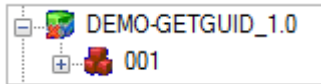
Durch das Suchen-Icon in der Symbolleiste oder über das Kontextmenü in der Konflikthanzeige klappt das Suchen-Fenster auf. Hierüber lassen sich alle Inhalte in der Konflikthanzeige nach einem Suchtext durchsuchen. Die Suche erstreckt sich nicht auf mehrere Konfliktknoten, sondern ist auf die aktuell gewählte Konfliktkategorie beschränkt!



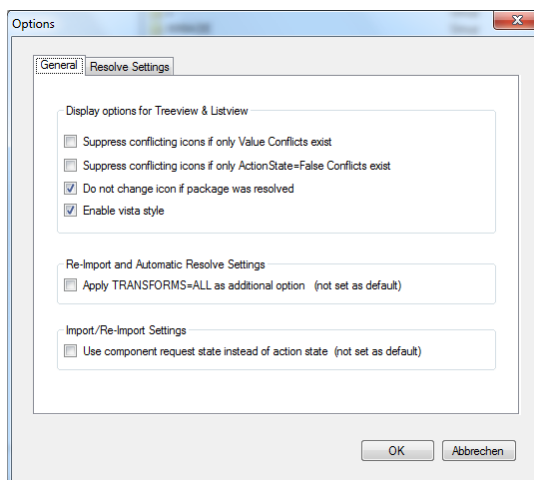
5.12 Optionen in der Benutzeroberfläche des Conflict Explorers

5.12.1 Register General

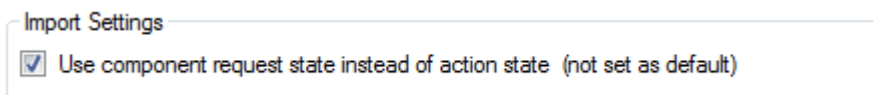
Über die Optionen, im *General* Register, zu finden in *Tools/Options*, lässt sich das Erscheinungsbild des *Family*- und *Package*-Icons im Treeview einstellen, so dass dieses bei *Value Conflicts* oder bei Konflikten aus Komponenten, die mit dem *ActionState=False* markiert sind, ohne Konflikt dargestellt werden.



Diese Option kann durch jeden Benutzer nach persönlichen Vorlieben eingestellt werden. Zudem wird über den selben Dialog angegeben, ob bei einem *Reimport* des Softwarepakets über das Icon in der Symbolliste oder über das Menü *Package/Re-import* alle im Paketverzeichnis befindlichen MST-Dateien angewendet werden sollen (*Re-import Settings*)



Die Option *Use component request state instead of action state* ermöglicht, dass beim Importierprozess nicht der *Aktionsstatus* der Komponenten ermittelt wird, sondern der *Anforderungsstatus*.



Ist die Option markiert, wird geprüft, ob eine Komponente **zur Installation vorgesehen ist**. Ist sie hingegen nicht markiert, dann wird geprüft, ob die Komponente durch Windows **installiert würde** (Standardeinstellung). Unterschiede ergeben sich vor allem, wenn die Komponente konditioniert ist und die *Condition* zudem *False* ergibt. Weitere Unterschiede findet man, wenn die Komponente in einer höheren Version (*KeyPath*) installiert ist oder die Komponente mit dem Attribut [msidbComponentAttributesNeverOverwrite](#) versehen ist und die dem *KeyPath* zugrundeliegenden Ressource bereits auf dem Computer installiert ist.

Die Einstellung kann sinnvoll sein, wenn man den Importprozess auf Computer ausübt, welche bereits mit diversen Anwendungsinstallationen vorbelastet sind, wenn der Importprozess also nicht auf einer *Clean Machine* ausgeführt wird. Wird diese Einstellung verwendet, erscheint auch keine Meldung der nachfolgenden Art beim Importieren der Softwarepakete.



Achtung: Die Ermittlung des *Aktionsstatus* (Standardeinstellung oder *ActionStateToRequestState=False*) ist auf jeden Fall die sicherere Methode, wenn es darum geht, zu ermitteln, ob eine Komponente effektiv installiert wird oder nicht. Daher sollten Sie diese Einstellung nur in Ausnahmefällen aktivieren. Auf der anderen Seite erkaufte man sich mit der markierten Option im schlechtesten Fall, dass der *Conflict Explorer* bei einer automatischen Korrektur gewisse Konflikte zuviel korrigiert, deren Behebung also eigentlich gar nicht nötig wäre. Das dem *Conflict Explorer* zugrundeliegende Prinzip „Korrigiere soviel wie nötig, aber so wenig wie möglich!“ wäre hier also nicht 100%ig umgesetzt.

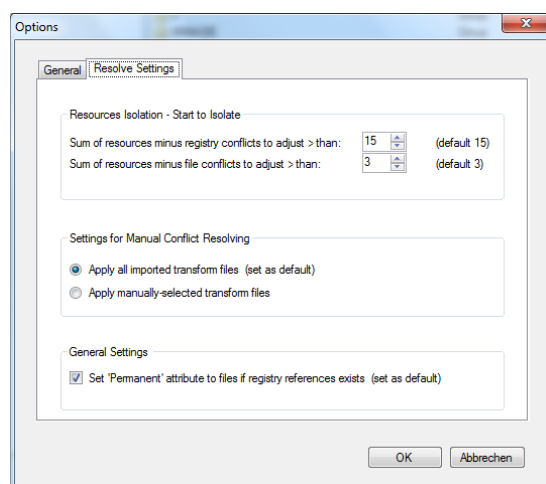
Die Einstellung kann auch über die *Settings.INI* dauerhaft eingestellt werden. Siehe [4.2.4 Settings.INI](#).
Achtung: Die Einstellung hat keinen Einfluss auf die Spaltenbezeichnung im *Conflict Explorer*!

ActionState

5.12.2 Register Resolve Settings

Über das Register *Resolve Settings* können für bestimmte Fälle die Standardeinstellungen für die Konfliktkorrektur verändert werden. Die Standardeinstellungen für die Ressourcenisolation gewähren einen Kompromiss, dass erst dann eine Isolation erfolgt, wenn die den Konflikten zugrundeliegenden Komponenten wesentlich mehr Ressourcen enthalten, als die Summe der konfliktierenden Ressourcen solcher Komponenten. Durch eine Erhöhung der Werte wird die Anzahl der Manipulationen des *Conflict Explorers* zur Zielerreichung reduziert, aber die Wahrscheinlichkeit, dass Anpassungen an Ressourcen vorgenommen werden, die eigentlich keine Anpassung nötig hätten, erhöht. Im Falle einer Verringerung der Grenzwerte verhält sich der *Conflict Explorer* dementsprechend gegenteilig.

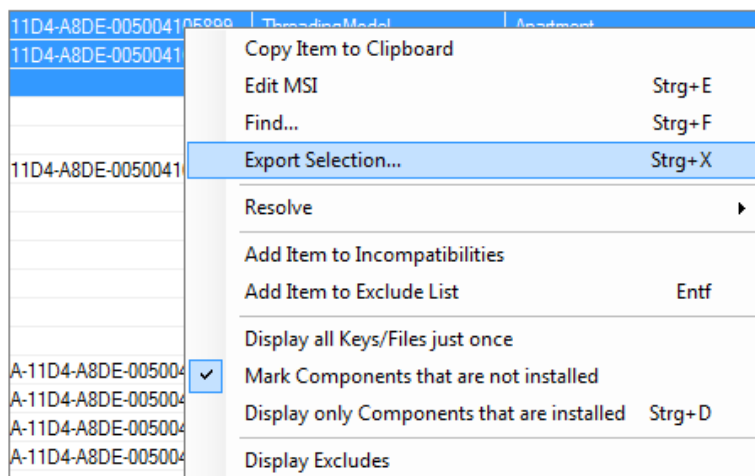
Die Einstellung unter *Settings for Manual Conflict Resolving* bestimmt den Umgang mit MST-Dateien bei einer automatischen Konfliktkorrektur. Mit der Option *Apply all imported transform files* wendet der *Conflict Explorer* automatisch alle früher importierten MST-Dateien an, bevor er die Konflikte korrigiert. Mit der zweiten Option erscheint bei der manuellen Konfliktkorrektur ein weiteres Dialogfenster, wo man die MST-Dateien auswählen kann, die vor der Korrektur angewendet werden. **Es sollten immer die gleichen Transformationen bei der Konfliktkorrektur angewendet werden, wie beim Importprozess!**



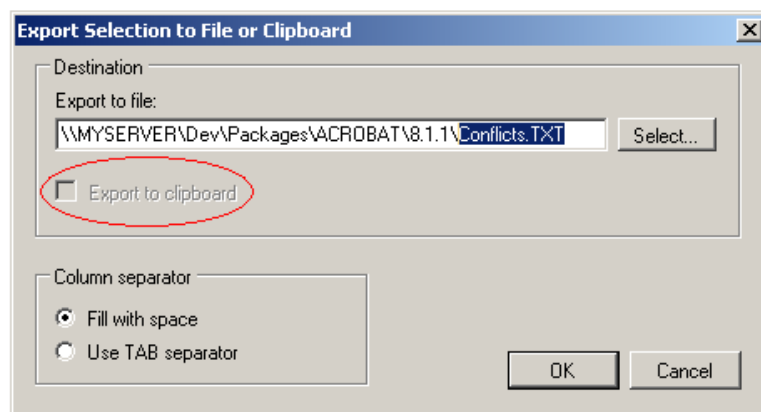
5.13 Konflikte exportieren

Nach einer Konfliktkorrektur wird üblicherweise die Ansicht der Konflikte aktualisiert und Konflikte verschwinden aus der Anzeige. Manchmal wäre es aber interessant zu wissen, was für Konflikte vor einer Konfliktkorrektur vorliegend waren oder wenn in einem Softwarebereitstellungsprozess zur Kontrolle das 4 Augen Prinzip durch eine weitere Person angewendet wird, möchte dieser vielleicht wissen, was für Konfliktausprägungen beim unveränderten Setup vor den Manipulationen des Software-Paketierers existierten.

Zu diesem Zweck ist es möglich, in der Konfliktanzeige, die entsprechenden Zeilen zu markieren, die man benötigt. Wählen Sie das Kontextmenü *Export Selection...*



Nun klappt ein Dialogfeld auf, wo Sie einen Dateinamen wählen können. Alternativ dazu können Sie mit der Option *Export to clipboard* die Ausgabe in die Zwischenablage umleiten.



5.14 Pakete verschieben, kopieren und löschen, sowie Gruppen erstellen

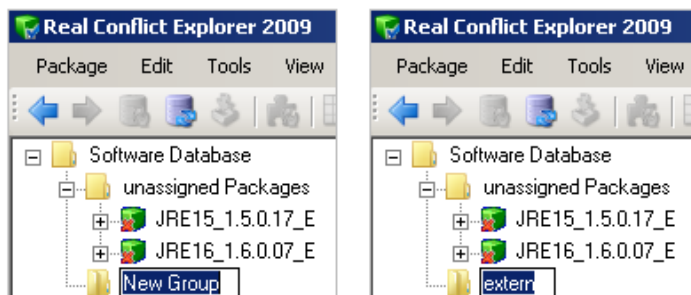
Der *Conflict Explorer* erlaubt ein einfaches Organisationsmanagement. Softwarepakete können mit wenigen Handgriffen in andere *Gruppen* verschoben oder kopiert werden. Löschaktionen sind genauso simpel durchführbar. Die Bedienung lehnt sich auch hier an den *Windows Explorer* an.

5.14.1 Gruppe erstellen oder umbenennen

Durch Klick auf die rechte Maustaste beim Knoten *Software Database* klappt ein Kontextmenü auf, welches Ihnen erlaubt, eine neue *Gruppe* zu erstellen.



Geben Sie anschliessend einen eigenen Name ein.

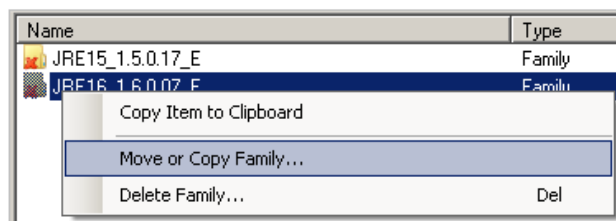


Wenn Sie zweimal auf einen bestehenden Gruppennamen klicken (kein Doppelklick), dann können Sie bestehende *Gruppen* umbenennen. Das Erscheinungsbild ist dann auch wie im letzten Bild abgebildet.

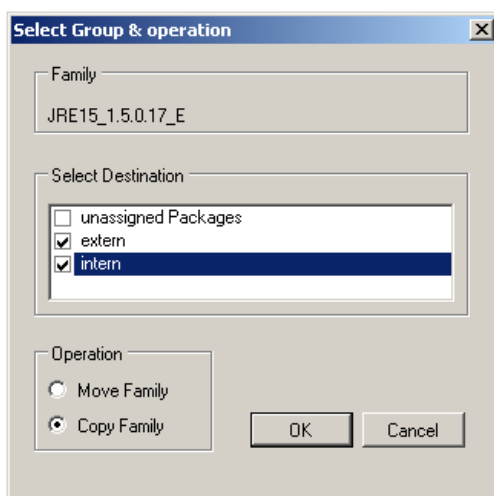
5.14.2 Paketfamilien verschieben oder kopieren

Verschieben und Kopieren lassen sich die Softwarepaketfamilien mit der Maus mittels *drag and drop*. Einfach mit der linken Maustaste auf den Familiennamen klicken, auf die *Gruppe* ziehen und die Maus loslassen. Wenn man die Wahl im Listview trifft, dann können gleichzeitig mehrere Pakete selektiert und verschoben oder kopiert werden. Mittels *drag and drop* bewirkt das gleichzeitige Drücken der *Ctrl*-Taste, dass die Verknüpfungen kopiert werden. Ohne Tastendruck werden sie verschoben.

Die gleiche Aufgabe wird auch durch das Kontextmenü erreicht. Das Kontextmenü durch Rechtsklick auf einen Familiennamen aufrufen...



Und danach die Operationswahl treffen. Hier können Kopier- und Verschiebeoperationen auf mehrere *Gruppen* angewendet werden.



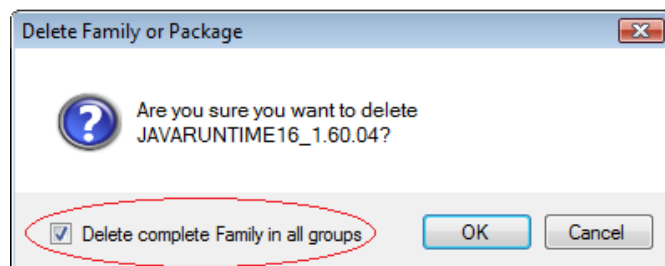
Hinweis:

Es lassen sich nur komplette *Families* kopieren und verschieben. Dazugehörige *Packages* kopiert oder verschiebt der *Conflict Explorer* automatisch mit!

Beachten Sie, dass Kopier- und Verschiebeoperationen Einfluss auf die Konflikte haben. Sie sollten nach solchen Operationen und vor allfälligen Interpretationen oder Konfliktkorrekturen ein *Rescan* der fokussierten Pakete vornehmen.

5.14.3 Paketfamilien löschen

Wählen Sie ein oder mehrere Softwarepaketfamilien aus und drücken Sie die -Taste. Folgender Dialog erscheint zur Bestätigung:



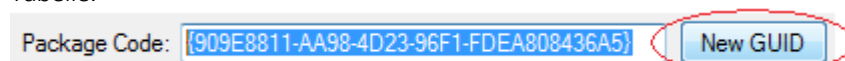
Sollte die Paketfamilie/n in mehreren Gruppen zu finden sein, dann wird die Checkbox *Delete complete Family in all groups* aktiviert. Durch die Wahl dieser Checkbox kann die Paketfamilie aus allen referenzierten Gruppen gelöscht werden.

Hinweis:

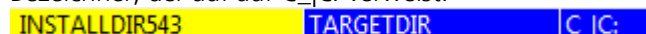
Das Löschen, Kopieren und Verschieben von Paketfamilien ist eine heikle Aufgabe und sollte in einem Team nur vom Administrator durchgeführt werden. Nach der Installation des *Conflict Explorers* ist nur die Person berechtigt, solche Manipulationen auszuführen, welche die Installation ausgeführt hat. Berechtigen Sie allfällig weitere Personen über den *AllowedToDelete*-Eintrag in der *Settings.INI* (siehe auch 4.2.4 *Settings.INI*)

5.15 Konflikte gegenüber dem Betriebssystem

Der *Conflict Explorer* kann auch dazu verwendet werden, um bei Softwarepaketen Konflikte zu ermitteln, die mit dem Betriebssystem entstehen können. Zurzeit können zwar die Betriebssystem-Ressourcen noch nicht direkt mit dem *Conflict Explorer* ermittelt und in die *Conflict Explorer* Datenbank importiert werden. Dies ist in einem späteren Release vorgesehen. Trotzdem müssen Sie auf diese Funktion nicht verzichten. Erstellen Sie in *Wise Package Studio* einen sogenannten *SOE-Snapshot* vom Betriebssystem (*Clean Machine*) mit allen aktuellen *Servicepacks* und *Patches*, aber ohne installierter Software. Diese Datei benennen Sie in eine *.MSI*-Datei um und editieren diese mit *Orca*. Erstellen Sie dort im *Summary Information* Dialog einen *PackageCode*, und löschen die *SelfReg* Tabelle:



Danach löschen Sie aus der *Directory Tabelle*, sofern vorhanden, einen allfälligen Directory-Bezeichner, der auf auf *C:\C:* verweist:

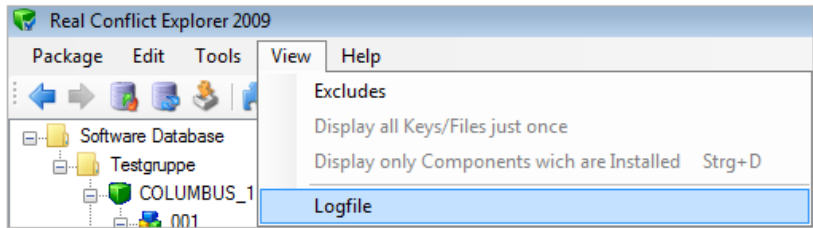


Nach dem Speichern der Datei importieren sie diese im *Conflict Explorer* wie ein gewöhnliches Softwarepaket. Ermitteln Sie die Konflikte und Sie ersehen dann die Konflikte des Betriebssystems aus dem Knoten des soeben importierten „Betriebssystempakets“. Dementsprechend finden Sie in den zum Betriebssystem konfliktierenden Softwarepaketen deren Konflikte, die Sie einfach mit einem *Automatic Resolve Conflicts* korrigieren können. Beachten Sie, dass Sie keine Konfliktkorrekturen am Betriebssystempaket vornehmen sollten. Es sind jeweils nur die Softwarepakete zu korrigieren und nicht das Betriebssystem!

5.16 Protokolldatei

Vor allem bei der Verwendung des *Conflict Explorers* in einem Team, können manchmal Fragen entstehen, welches Teammitglied wann welche Operation durchgeführt hat. Solche Fragen kann ein Blick in die Protokolldatei beantworten.

Staten Sie *View/Logfile*

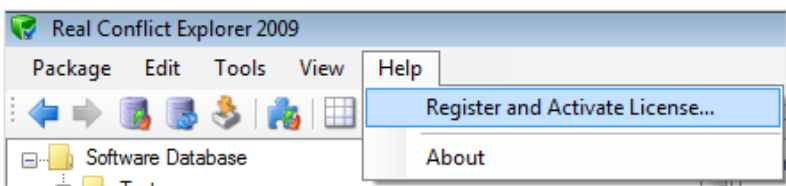


Worauf die Informationen im rechten Fenster angezeigt werden.

Time	User	Task	Package	Speed
11.12.2008 14:04:27	oberlind	Delete Family	COLUMBUS_1.8.2_E	
10.12.2008 14:50:38	oberlind	Scan all Packages		2sec
10.12.2008 14:49:58	oberlind	Import Package	COLUMBUS_1.8.2_E_001	18sec
10.12.2008 14:44:04	oberlind	Delete Family	COLUMBUS_1.8.2_E	
10.12.2008 14:43:46	oberlind	Import Package	COLUMBUS_1.8.2_E_001	9sec
10.12.2008 14:43:41	oberlind	Delete Family	MSREPORTVIEW_8.0_E	
10.12.2008 14:43:06	oberlind	Delete Family	LMS-EXCELPRINT_1.0_E	
10.12.2008 14:16:57	oberlind	Import Package	EXPRESSO_3.0_E_001	11sec
10.12.2008 14:15:51	oberlind	Import Package	GUICC_1.0_E_001	17sec
10.12.2008 14:14:07	oberlind	Import Package	LMS-EXCELPRINT_1.0_E_001	59sec
10.12.2008 13:59:57	oberlind	Import Package	MSREPORTVIEW_8.0_E_001	14min 5sec
10.12.2008 13:59:25	oberlind	Import Package	NETVIZGUI_7.5_E_001	16sec
10.12.2008 13:58:18	oberlind	Import Package	PASSPORT-PC-TO-HOST_2.00.7_E_001	13sec
10.12.2008 13:56:24	oberlind	Import Package	PASSWORD-SAFE_3.13_E_001	50sec
10.12.2008 13:55:34	oberlind	Import Package	PF-HOUSEKEEPING_2.2_E_001	28sec
10.12.2008 13:53:51	oberlind	Import Package	PF-SCANNAPPLICATION_5.0.5_E_001	1min 22sec
10.12.2008 13:53:09	oberlind	Import Package	PKS-PLOG_4.3_E_001	11sec
10.12.2008 13:52:22	oberlind	Import Package	PPTVIEW2007_12.0_E_001	20sec
10.12.2008 13:51:13	oberlind	Import Package	SAFESIGN_3.0.11_E_001	18sec
10.12.2008 13:50:24	oberlind	Import Package	SCANDALL_21_E_001	18sec
10.12.2008 13:49:10	oberlind	Import Package	SENTINEL-SYSTEM-DRIVER_5.39.2_E_001	7sec
10.12.2008 13:46:13	oberlind	Import Package	TRANSLATOR_1.2_E_001	17sec

5.17 Lizenz-Aktivierungsprozess

Der *Conflict Explorer 2009* steht nur als Download in Form einer Evaluationsversion bereit. Diese Evaluationsversion unterliegt gewissen Beschränkungen. Entscheidet man sich für den Erwerb des *Conflict Explorers 2009*, so wählt man auf einem Computer mit installiertem Mail-Client *Register and Activate License...*



Wählen Sie zunächst bei wem Sie bestellen möchten. Unter *Seller Information* können Sie das Produkt bei einem unserer Vertragspartnern, bzw. ihrem Ansprechpartner für den *Conflict Explorer*

kaufen. Hier müssen Sie im Besitz dessen E-Mail-Adresse sein. Oder Sie erwerben das Produkt direkt bei uns: Dann belassen Sie die Einstellung wie sie gewählt ist.

Danach wählen Sie unter *Order Information*, welchen Leistungsumfang Sie käuflich erwerben möchten. Die Preisgestaltung ersehen Sie auf der Homepage unter www.realpackaging.net.

Denken Sie daran, dass Sie nicht zu knapp kalkulieren. Wenn Sie beispielsweise 90 Softwarepakete im Einsatz haben, ist es zukunftssicherer und kommt es Ihnen günstiger, wenn Sie einen höheren Leistungslevel als *BASIC* erwerben. Haben Sie das Produkt aber auf einen kleinen Leistungslevel aktiviert, können Sie später auch auf einen höheren Leistungslevel aktivieren.

Mit der Schaltfläche *Order activation by mail* schliessen Sie Ihren Auftrag ab. Dies löst ein E-Mail aus, worauf Sie ihre Rechnung mit Angabe des Rechnungsbetrages, weiteren Informationen und der Bankverbindung erhalten. Sobald wir Ihren Zahlungseingang verzeichnen, werden wir Ihnen ein *Aktivierungs-pack* zusenden, welches zu installieren ist. Danach ist ihr Produkt für den entsprechenden Leistungslevel freigeschaltet.

Register & Activation

Select activation options

Seller Information

I want to activate Conflict Explorer by reseller (mail to reseller)

I want to activate Conflict Explorer by RealPackaging.net (mail to us)

Order Information

EVALUATION (free for max. 30 package imports, 60 days)

BASIC (max. 100 package imports - fee required)

PROFESSIONAL (max. 200 package imports - fee required)

ENTERPRISE (unlimited package imports - fee required)

Currently activated to EVALUATION feature set.

Serial key (send print screen from this dialog if mail activation does not work)

REROL PXE01 TCILF NOC02 MC-AT SIV02 02

activation@realpackaging.net

Order activation by Mail Cancel

Hinweis:

Sollten Sie das Produkt später auf eine neue Plattform zügeln, dann wird der *Conflict Explorer* unbenutzbar. Für solche Fälle können Sie den *Conflict Explorer* einfach reaktivieren. Eine Reaktivierung ist kostenlos und wird in der Regel binnen 24h durchgeführt. Führen Sie eine Reaktivierung mit dem gleichen E-Mail-Account durch, wie Sie die ursprüngliche Aktivierung vorgenommen haben, dies erleichtert uns das Handling.

Hinweis:

Achten Sie darauf, dass Sie eine Aktivierung erst dann ausführen, wenn Sie die Evaluationsversion erfolgreich auf der zum Einsatz vorgesehenen Zielplattform installiert haben. Änderungen am Server oder an der Verzeichnisstruktur können nachher nur noch mit einer verbundenen Reaktivierung vorgenommen werden!

Eine Produktaktivierung ist in der Regel einmal durchzuführen und alle Teammitglieder der Unternehmung sind danach berechtigt, im Netzwerkverbund dieses Produkt frei zu verwenden.

5.18 Kurztasten

- Ctrl-A** Im Konfliktanzeigefenster führt Ctrl-A zur Selektion aller Konflikte mit *ActionState=True*. Wird danach noch einmal Ctrl-A gewählt, so werden alle Einträge selektiert.
- Del** Im Konfliktanzeigefenster führt dazu, dass der Konflikt in die *Excludeliste* aufgenommen wird. Aus der *Excludeliste* wird der entfernte Konflikt für die Standardansicht wieder verfügbar. Auf einem *Package* löscht es ein *Package*, auf einem *Familyname* löscht es die *Family*, samt allen *Packages* und auf einer *Gruppe* löscht es die komplette *Gruppe*.
- F5** Bei aktiviertem Treeview (linker Fensterteil) oder Listview (rechter Fensterteil) führt F5 zu einem Refresh und neuem Einlesen der Paketdaten.
- Backspace** Im Treeview und Listview wählt dies den übergeordneten Knoten.
- F3** Im Konfliktanzeigefenster führt F3 zum Suchen des nächsten Vorkommens des letzt eingegebenen Suchtextes.
- F8** Bei geöffneter Gruppe wird die nächste importierte Family (gelb) selektiert
- Ctrl** Kurz vor dem Aufruf von *Edit MSI* aufgerufen und nach dem Start der Anwendung losgelassen, kopiert dies den Pfad der MSI-Datei in die Zwischenablage. Dies kann beispielsweise verwendet werden, wenn man *Orca* startet und nachher eine Transformation anwenden will. Oft ist im darauffolgenden Dialog noch das letzte gewählte Verzeichnis aktiv, was vielfach unerwünscht ist. Hier in der Eingabe des Dateinamens einfach Shift-Insert und <Enter> wählen, damit man sogleich auf den Pfad Zugriff hat.

Alle anderen Kurztasten sind im Menü oder im Kontextmenü dokumentiert.

5.19 Häufige Fragen

- F: Kann ich mit dem *Conflict Explorer 2009* auch eine MSP-Patchdatei importieren?
- A: Nein. Ein Patch kann aber mittels einem *In-Place-Update* auf die Basisinstallation (MSI) angewendet werden und diese kann anschliessend importiert werden. (*msiexec /p /a*)
-
- F: Wie werden Ressourcen behandelt, die in *CustomActions* untergebracht sind?
- A: Die Ausführung von *CustomActions* zählt für den *Windows Installer* unbestritten zu den komplexesten und kritischsten Vorgängen im Installationsprozess. Die Inhalte von *CustomActions* bleiben selbst dem *Windows Installer* verborgen, daher kann auch der *Conflict Explorer* diese Ressourcen nicht ermitteln. *CustomActions* sollten nach Best-Practice Regeln immer datengesteuert implementiert werden. D.h., dass *CustomActions* nicht direkt Ressourcen manipulieren sollten, sondern nur indirekt. In einem solchen Szenario und wenn die nötigen Ressourcen in den Datenbanktabellen vorliegend sind, kann der *Conflict Explorer* auf alle Ressourcen zugreifen.
- Mit dem *Conflict Explorer* müssen wir aber manchmal auch Kompromisse eingestehen. Werden tatsächlich einige Ressourcen über *CustomActions* direkt implementiert, so werden diese nunmal nicht berücksichtigt. Diese meist kleine Menge an Ressourcen steht oft in keinem Verhältnis zu der Anzahl an Ressourcen, die durch den *Conflict Explorer* berücksichtigt werden.
-
- F: Warum kann man *Pakettfamilien* und *Packages* nicht umbenennen?
- A: Auf der Namensgebung beim Importieren von Softwarepaketen basieren verschiedene nachfolgende Aufgaben des *Conflict Explorers*. Nicht nur, dass die *Packagefamily-Excludeliste* auf bestehende Namen zurückgreift und andere Konfliktanzeigen auf

existierende Namen verweisen. Es sind auch Transaktionen wie ein *ReImport* mit der Namensgebung verbunden. Wenn jetzt jemand ein Softwarepaket umbenennen würde, dessen Namensgebung ursprünglich beispielsweise aus dem Einlesen von Properties aus der *Property*-Tabelle der MSI-Datei entstanden ist, wie sollte sich dann der Reimportprozess des umbenannten Pakets vollziehen? Soll es das bestehende umbenannte Paket löschen und die Originalbezeichnung (aus der *Property*-Tabelle ausgelesen) wieder herstellen? Oder sollte es die *Property*-Tabelle einfach ignorieren?

Ich denke, dass beide Varianten zu einer Verwirrung des Benutzers führen würden. Daher wurde im *Conflict Explorer* bewusst auf diese Funktion verzichtet.

Sollten Pakete andere Namen bekommen, so sind diese Pakete mit anderen Namen neu zu importieren und die *Packages* mit den alten Namensbezeichnungen kann man schliesslich aus dem *Conflict Explorer* löschen.

F: Es wurde vor einem halben Jahr ein Softwarepaket importiert und automatisch korrigiert. Damals war es im Anschluss konfliktfrei. Dieses konfliktfreie Softwarepaket wurde in die Produktion überführt und auf die Computer im Unternehmen installiert.

Unterdessen sind weitere neue Softwarepakete importiert worden und das alte Softwarepaket enthält im *Conflict Explorer* nun neue Konflikte. Warum ist dies so, und wie soll darauf reagiert werden?

A: Durch ein Software Konfliktmanagement geht man in der Regel immer gewisse Kompromisse ein. Da nun in diesem halben Jahr neue Softwarepakete importiert wurden, ist es tatsächlich so, dass nun das alte Softwareprodukt neue Konflikte gegenüber den neuen Softwareprodukten aufweisen kann. Ob und wie in einem Unternehmen darauf reagiert wird, kann im Einzelfall unterschiedlich sein. Eine Möglichkeit ist die, dass man, wenn es sich um schlimme Konflikte handelt, erneut eine Konfliktkorrektur initiiert und aus Transparenzgründen am besten in einer neuen Transformationsdatei abspeichert. Für Neuinstallationen wäre dann das neu korrigierte Softwarepaket zu verwenden.

Vielleicht verhindern Unternehmensrichtlinien deren Nachkorrektur und eine Anpassung wäre dann erst mit einem Update oder einem neuen Release vorzunehmen. Auch für einen solchen Fall hat der *Conflict Explorer* durchaus seine Daseinsberechtigung. Besser ein um ev. tausende Konflikte bereinigtes Softwarepaket einsetzen (Erstimplementation), als gar kein Konfliktmanagement anzuwenden.

F: Ich habe ein Softwarepaket X importiert Dieses hat viele Konflikte gegenüber Softwarepaket Y. Warum hat Softwarepaket Y dann keine Konflikte gegenüber Softwarepaket X?

A: Konflikte kommen zwar im Verbund von Software zustande, trotzdem bedeutet dies nicht, dass automatisch die Konflikte gegenseitig sind. Im besagten Fall bedeutet dies, dass im zweiten Softwarepaket Y, die zugrundeliegenden Komponenten über die Attribute richtig markiert sind oder schon eine Konfliktkorrektur vollzogen wurde, im anderen Softwarepaket X hingegen nicht. *Value Conflicts* sind allerdings gegenseitig. Das heisst, wenn ein *Value Conflict* im einen Softwareprodukt ausgewiesen wird, dann muss der entsprechende *Value Conflict* auch im konfliktierenden Softwarepaket angezeigt werden, ausser man hätte diesen Konflikt in die *Excludelist* übernommen oder die *Packagefamily-Excludeliste* wurde erweitert.

F: Testweise habe ich ein Softwarepaket zweimal importiert. Einmal mit einem anderen Namen. Warum werden mir nach einem *Scan Package* keine Konflikte ausgewiesen, auch wenn die Pakete in der gleichen Gruppe sind?

A: Da auch die *ComponentId's* in der Component-Tabelle der zugrundeliegenden MSI-Datei unverändert sind und keine Veränderungen appliziert wurden, aufgrund denen ein *Value Conflict* hätte entstehen könnte, sind keine Konflikte vorhanden. Der *Conflict Explorer* reagiert richtig.

6. Kommandozeilenoptionen

ImportMsi.EXE und *ConflictExplorer.EXE* können mit Kommandozeilenoptionen gesteuert werden. Durch den Einsatz von Kommandozeilenoptionen sind automatisierte Prozessabfolgen realisierbar. Insbesondere ist so eine elegante Integration in die *Projekte* von *Wise Package Studio* realisierbar.

Kommandozeilenoptionen für *ConflictExplorer.EXE*

Kommandozeilenoption	Beschreibung
FAMILYNAME PACKAGENAME Beispiel: ConflictExplorer.EXE ADOBEREADER_9.0 001	Öffnet den <i>Conflict Explorer 2009</i> im gewählten Knoten. Ein Navigieren zum Paket ist durch den Softwarepaketierer nicht mehr erforderlich. Es wird der erstfundene Eintrag in der erstmöglichen Gruppe ausgewählt.

Kommandozeilenoptionen für *ImportMsi.EXE*

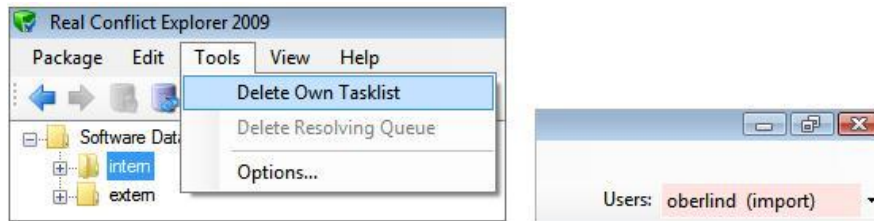
Kommandozeilenoption	Beschreibung
package.msi Beispiel: ImportMsi.exe package.msi	MSI-Datei, die zu importieren ist. Wird die angegebene Datei nicht gefunden oder nicht angegeben, dann öffnet der Dateidialog-Wizzard.
/S/Q Beispiel: ImportMsi.exe package.msi /S	Führt die Aktion ohne Benutzerinteraktion aus (sofern Family- und Packagenamen ermittelbar sind). Insbesondere werden keine Fehlermeldungen angezeigt. Diese Option ist bei einem automatisierten Multipaketimport über ein Script empfohlen.
/M Beispiel: ImportMsi.exe package.msi /M	Ignoriert die Prüfung, ob ein Softwarepaket vor der Installation bereits auf dem Computer installiert ist und ob der Computer der Richtlinie einer <i>Clean Machine</i> entspricht (siehe <i>Clean-MachinePackageLimitValue</i> in 4.2.4 Settings.INI). Entsprechende Fehlermeldungen erscheinen nicht.
/A Beispiel: ImportMsi.exe package.msi /A	Ermittelt beim Importieren den Anforderungsstatus der Komponenten, anstatt den Aktionsstatus (siehe <i>ActionStateToRequestState</i> in 4.2.4 Settings.INI).
/E Beispiel: ImportMsi.exe package.msi FAMILY=ADOBEREADER_9.9 PACKAGE=001 /E	Ermöglicht bei einem Multipaketimport via Import-Script, dass nur Softwarepakete importiert werden, die nicht schon in der Datenbank vorhanden sind. <i>ImportMsi.EXE</i> beendet die Ausführung somit ohne Funktion, wenn das mittels <i>FAMILY=xxx</i> und <i>PACKAGE=yyy</i> deklarierte Paket bereits importiert wurde. <i>/E</i> funktioniert nur zusammen mit <i>FAMILY=xx</i> und <i>PACKAGE=yyy</i> .

<p>/P</p>	<p>Verhindert die Ermittlung der SelfReg-Einträge und führt beim Importprozess auch keine Pseudo Windows Installer Installation des Produktes aus.</p>
<p>/?</p>	<p>Zeigt die wichtigsten Kommandozeilenoptionen</p>
<p>PROPERTY=Value Beispiel: ImportMsi.exe package.msi ALLUSERS=1</p>	<p>Windows Installer Properties können mit der Befehlszeile mitgegeben werden. (Dies kann bei der Ausführung über ein Import-Script erforderlich sein)</p>
<p>TRANSFORMS=Name[;Name[;ALL]] Beispiel: ImportMsi.exe package.msi TRANSFORMS=ALL</p>	<p>Wendet beim Importierprozess die ausgewiesenen Transformationen an. Ist die Anwendung mehrerer Transformationen erforderlich, so verwenden Sie zwischen den einzelnen Transformationen ein Semikolon „;“. Verwenden Sie das Schlüsselwort ALL, wenn alle Transformationen im Verzeichnis der MSI-Datei angewendet werden sollen. Schliessen Sie die Transformationsdateien in Anführungszeichen, wenn Leerzeichen im Text enthalten sind.</p>
<p>FAMILY=FAMILYNAME Beispiel: ImportMsi.exe FAMILY=ADOBEREADER_9.0</p>	<p>Importiert das Softwarepaket mit dem angegebenen Familiennamen. Sind <i>Familyname</i> und <i>Packagename</i> durch Kommandozeilenoptionen oder indirekt über die <i>Property</i>-Tabelle angegeben, erscheint kein Dialog, wo die Namen angegeben werden müssen.</p>
<p>PACKAGE=PACKAGENAME Beispiel: ImportMsi.exe FAMILY=ADOBEREADER_9.0 PACKAGE=001</p>	<p>Importiert das Softwarepaket mit dem angegebenen Paketnamen. Sind <i>Familyname</i> und <i>Packagename</i> durch Kommandozeilenoptionen oder indirekt über die <i>Property</i>-Tabelle angegeben, erscheint kein Dialog, wo die Namen angegeben werden müssen.</p>
<p>GROUPS=Name[;Name[;...]] Beispiel: ImportMsi.exe package.msi GROUPS=German</p>	<p>Geben Sie hier die Gruppennamen an, mit welchen das Softwarepaket verknüpft werden soll. Mehrere Gruppennamen trennen Sie mit einem Semikolon „;“. Werden die Gruppennamen nicht gefunden oder nicht über diese Kommandozeilenoption angegeben, so wird das Softwarepaket der Gruppe ‚<i>unassigned Packages</i>‘ zugewiesen.</p>
<p>PROPERTYFAMILY=PROPERTYNAME PROPERTYPACKAGE=PACKAGENAME Beispiel: ImportMsi.exe package.msi PROPERTYFAMILY=CEFAMILY PROPERTYPACKAGE=CEPACKAGE</p>	<p>Verwenden Sie diese Option, wenn Sie in der <i>Property</i>-Tabelle <i>Eigenschaften</i> für die Zuweisung der Namen verwenden. In einem automatisierten Paketierungsprozess ist dies die empfohlene Variante.</p>

7. Wichtige, bisher unbehandelte Funktionen

7.1 Delete Own Tasklist

Mit *Delete Own Tasklist* aus dem Menü *Tools* können alle eigenen Tasks gelöscht werden, damit diese keine anderen Prozesse blockieren. Dies ist eigentlich nur dann nötig, wenn Prozesse wie *ImportMsi* nicht bis zu Ende ausgeführt werden.



Conflict Explorer 2009 erlaubt nur das Löschen der eigenen Tasks.

7.2 Hilfe

Die Hilfedokumente sind im Verzeichnis *Help* abgelegt. Ist ein PDF Reader installiert und stehen die Ländereinstellungen auf Deutsch oder Englisch, so kann dieses Dokument auch mit dem Menü *Help/Help...* oder mit F1 aktiviert werden.

8. Stichwortverzeichnis

%

%WINDIR%\System32..... 27

.

.NET Berechtigungen 8
 .NET Framework 8

A

Abhängigkeiten 7, 8
 Ablagepfade 10
 ACL
 Berechtigungseinschränkungen 13
 ActionState 7, 11, 19, 23
 ActionStateToRequestState 12, 30
 Add Item to Excludelist 20, 21
 Add Item to Incompatibilities 20
 AddFullTrustingAndShortcuts.vbs 8
 Administrator 13, 33
 Administratorrechte 8
 Aktionsstatus 7, 12, 29
 Aktivierung
 Lizenz 34
 Aktivierungspack 35
 Align ComponentId 24
 AllowedToDelete 13, 33
 Anforderungsstatus 12, 29
 Ansicht
 ausschliessen 21
 Applikations-Lebenszyklus 4
 Apply all transforms from package directory 15
 AppSearch-Properties 7
 Arbeitserleichterungen 7
 ausschliessen
 Ansicht 21
 Automatische Konflikthanpassung 14, 26

B

Bad packages 25
 BASIC 35
 Benutzer
 Berechtigung 13
 Berechtigung 33
 Best-Practice Regeln 18

Betriebssystem 33
 Konflikte gegen 5
 blau, blaue Hintergrundfarbe 14, 19

C

CheckConflicts.EXE 8, 10
 Clean Machine 11, 13
 CleanMachinePackageLimitValue 11, 38
 Cleartext 7
 Client Setup 8
 COM-Dateien 18, 23
 COM-Komponenten 3
 ComponentId 25
 Anpassung der 24
 ComponentId's 37
 ComponentIsolationX 27
 Computertyp
 Set 21
 Condition 13
 Conflict Explorer
 Datenbank 15
 in einem Team 34
 reaktivieren 35
 ConflictExplorer.EXE 8
 Kommandozeilenoptionen 38
 Conflicts
 Knoten 20
 Conflicts Knoten 17
 ConponentId
 Abgleich 22
 content 4
 Create Transform 24
 Ctrl-A 36
 Ctrl-Taste 32
 CustomActions 7, 36
 datengesteuert 36

D

Dateikonflikt 23
 Dateipfad
 öffnen 14
 Dateiversion 3
 Dateiversionen 18
 Daten
 konsistente 7
 Datenkonsistenz 7
 Deinstallation 17
 Paket 19
 Deinstallation eines Softwarepakets 4

Del.....	20, 33, 36
Delete complete Family in all groups.....	33
Delete Own Tasklist	39
Delete Resolving Queue.....	24
Development	11
Display Excludes zu wählen.....	20
DLL-Datei.....	27
DllRegisterServer()	3
Domänenbenutzer	13
Don't import SelfReg informations	15
drag and drop	32

E

Einstellungen	
bleibende.....	19
Entwicklungsshare	11
Erscheinungsbild	
Icon.....	29
Ersetzpfade	10
Evaluationsversion.....	34
Excludelist	20, 27
EXE-Datei	27
Export Selection	31
Export to clipboard	31

F

Familiennamen	9
Families	
komplette	32
Family.....	6
FAMILY	39
Familyname.....	6, 15
FAMILYNAME	38
File	17
File Conflicts.....	19
File Ressourcen	27
File Versioning Rules	18
Filekonflikte	27
Files	7
Filteroperationen	19
Fragen	36
freischalteten	35
FullTrust	8

G

gelbes Icon	9, 17
globale Regeln.....	4
globaler Designfehler	4
Good packages.....	25
GROUPS	39

Gruppen/n	21
derselben	21
umbenennen	31
erstellen	13, 31
mehrere/n.....	32, 33
unterschiedliche.....	27

H

Hauptgruppen von Konflikten	4
heterogenes Desktopumfeld	5
Hilfe	40

I

ICE-Konflikte	4
Icon	
gelbes	17
Import	
performanter	7
robuster.....	7
Importdatum	7
Importieren von Softwarepaketen	15
Importierprozess	15
ImportMsi	8
Kommandozeilenoptionen.....	9
ImportMsi.EXE.....	8, 15
Kommandozeilenoptionen.....	38
Importprozess	
Team	9
Incompatibilities	14, 21
Gültigkeit	21
Inhalte	
Registrykey.....	18
In-Place-Update	36
Installation	
Paket	19
Installationsdatei	
Nichtvorfinden	10
Installationsstatus	
ungültiger	18
Integration/Test.....	11
Internal Consistency Evaluators	4
IsolationBeginFiles	12
IsolationBeginRegistry	11

K

Key.....	18
KeyPath.....	13, 27
Regeln	27
Kommandozeilenoptionen	9, 38, 39
ImportMsi	9

Komponente/n.....	12, 27
blau eingefärbte	23
Komponentendesigns	
fehlerhaftes	4
Komponenteninhalte	24
Komponentenisolation/en	11, 27
Konflikt	
einmal	19
mehrfach	19
Verhältnis	36
Konfliktanpassung	
automatische.....	14, 26
beschränken	24
erweitern	24
Konfliktanpassungen	
benutzerdefinierte.....	24
Konfliktanzeige/n	
erweiterte.....	7, 14
Konfliktarten	17
Konfliktbehebung.....	3
Konfliktbereinigung	
automatische.....	7
Standardregeln	27
Konflikte	
ausschliessen.....	20
ausfiltern	19
Ausschluss	19
bereinigen.....	21
echte.....	7
ermitteln.....	14
exportieren	30
gegenseitig	37
gegenüber Betriebssystem	33
gegenüber dem Betriebssystem	33
keine	27
neue.....	37
nicht bereinigt	18
Rest.....	23
selektive.....	21
Suche	28
zur Konfliktbereinigung vorgesehene	24
zwischen Packages.....	27
zwischen Ressourcen.....	27
konfliktfrei.....	37
konfliktierendes Paket	
gemeinsam	21
Konfliktkorrektur	
automatische.....	21
Ein zweites Mal.....	23
Konfliktlösung	3
Konfliktmanagement	3
Konfliktvalidierung.....	3
Kopieren	
Pakete.....	31
Kosten	4
Kurztasten.....	36
Backspace	36

Ctrl.....	36
Ctrl-A	36
Del	36
F3	36

L

Leistungslevel	35
LOCAL Isolation.....	18
lokale Regeln	4
lokales Design	3
Löschen	
Pakete	31

M

Manuelle Konfliktkorrektur	24
Menü	
Package/Reimport.....	29
Tools/Options.....	29
View/Logfile	34
Menü View.	19
MSI-Dateien	
Hersteller.....	7
msidbComponentAttributesNeverOverwrite	13
msidbComponentAttributesPermanent	23, 27
msidbComponentAttributesSharedDllRefCount	27
MST	
separate	7
MST-Datei	
bestehende	22
MST-Dateien.....	30
MultiPackageImport.vbs.....	15

N

Name	18
Namensgebung	
inkonsequente	18
neue Konflikte	37
Nichtfunktionieren	18
Nichtvorfinden der MSI-Datei.....	10

O

Optionen.....	29
Orca	7
OrcaPath.....	10
Order activation by mail	35
Order Information	35
Organisatorische Massnahmen	19

P

Package	6
Bezeichnung	9
Icon	29
PACKAGE	39
Package/Import	15, 17
Packagefamily	27
Packagefamily-Exclude.....	20
Packagefamily-Excludeliste	7
PACKAGENAME.....	38
Packages	
unassigned	9
Paket	
mit Konflikten	14
ohne Konflikte	14
Pakete	
kopieren	31
löschen	13, 31
verschieben	13, 31
Paketfamilien	6
kopieren	32
löschen	33
verschieben	32
Paketverzeichnis	29
Patchdatei	36
PathReplacements	10
PE-Dateien	8
Permanent Attribut.....	22
Plattform	
zügeln	35
Prepare Selected Conflicts to Resolve.....	24
proaktive Konfliktkorrektur	5
Production	11
Produktaktivierung	35
Produktion	11
ProjectDir	
Wise.....	8
Projekt	
Wise.....	8
Properties	7
Property	
Tabelle	9
PropertyFamily.....	9
PROPERTYFAMILY	39
PropertyPackage	9
PROPERTYPACKAGE	39
Propertytabelle	37
Protokolldatei	34
Prozesse	
blockieren	39
Prüf-Regeln	27
Pseudoinstallation	8, 15

Q

Queue	24
-------------	----

R

Registrierungsschlüssel	
gleiche	4
Registrierungsverweise	23
Registry.....	7, 10, 17
Registry Conflicts	18
Registry to Registry Conflicts	18
Registry to SelfReg.....	17
Registry to SelfReg Conflicts	18
Registry to SelfReg Value Conflicts	18
Registry Value Conflicts	18
Registrykey	18
Registry-Keypath	27
Reimport.....	12
Reimport Settings	29
Reimportieren	14
RequestState	13
Rescan	20
Resolve Settings.....	30
Ressourcen	
gemeinsame	18
isoliert	27
Ressourcenisolation	
Standardeinstellungen	30
Ressourcentyp	17
Restkonflikte.....	23
Root	18
roter Eintrag	24

S

Scan Package Conflicts.....	10, 17, 27
Select Transform to Edit or Create Now	22
SelfReg.....	7, 17
SelfReg to Registry Conflicts	18
SelfReg to Registry Value Conflicts	18
SelfReg to SelfReg Conflicts	18
SelfReg to SelfReg Value Conflicts	18
SelfReg-Inhalte	8
SelfReg-Konflikte	18
SelfReg-Registrierungen	18
SelfReg-Tabelle/n	8, 18
Seller Information.....	34
Semikolon	10
Server	
verschiedene	10
Settings	
globale.....	10

Settings for Automatic Resolve	30
Settings.INI	8, 10
ShareDll	22
Shares	10
Softwarebereitstellungsprozess.....	10, 30
Softwarekonflikte in Softwarepaketen	3
Softwarepakete	
robuste	3, 4
Standard-Ablauf	17
Standard-Regeln	27
Standardverzeichnis.....	10
Suche	14, 28
MSI Datei	10

T

Tasks	
löschen	39
Teammitglieder	35
Terminologie	6
Tragweite von Konflikten	5
Transformation	21, 24
Transformationen	
automatisch	12
Transformdatei	5
TRANSFORMS	39
TransformsAll	12
TransformsAllReimport	12

U

UAC	8
umbenennen	
Packages	36

Paketfamilien	36
unassigned Packages	9
Upgrade eines Softwarepakets	4
UpgradeCode	6
Userliste	14
Users	
Anzeige	9

V

Value Conflicts	17, 18, 21
Verbund von Software	37
Verschieben	
Pakete	31
Version	6
aktuellste	18
Verzeichnisstruktur	10
Vista	8

W

Warteschlange	24
Windows Installer Properties	7
Windows Vista	8
Wise Package Studio	
Projekte	38
Wise-Projekt	8

Z

Zugriff	
lesend	9