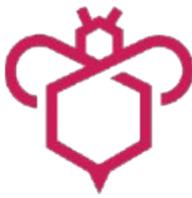


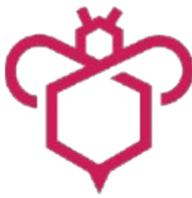


opsi-background-install

Installation während Software läuft



- Anlass
- Probleme
- Lösungsansätze
 - Vermeidung von Fehlern
 - Sichtbarkeit
- Anforderungen an die Implementierung



- Umsetzung der experimentellen Implementierung
- Vorführung
- Offene Fragen
- Roadmap



Anlass

- Der Wunsch Software bei eingelogtem user und (potentiell) laufender Software zu installieren kommt durch:
 - 24/7 Betrieb wie in Krankenhäusern
 - Keine Möglichkeit für WakeOnLan in Schulen
 - Nutzer loggen sich nie aus und lassen Rechner und Programme laufen.



Probleme

- Konflikte mit laufender Software
- Störung durch aufpopende Fenster



Problem

- Konflikte mit laufender Software



Probleme: Konflikte: Typen

Erhoben an 26 verschiedenen
Beispielen,
bei denen die Zielsoftware lief:

1. succ / killed

- Installation erfolgreich
- laufende Software 'abgeschossen'

2. failed / running

- Installation fehlgeschlagen
- laufende Software läuft weiter

3. succ / running / reboot

- Installation erfolgreich
- laufende Software läuft weiter
- Reboot angefordert

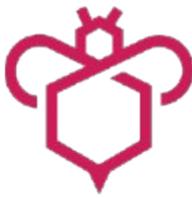
4. succ / killed dependent

- Installation erfolgreich
- abhängige Software 'abgeschossen'

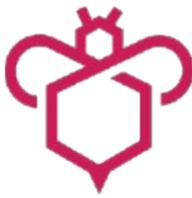
5. reboot while install

- Reboot während der Installation angefordert

Lösungsansätze: Vermeidung von Fehlern 1



- Eine Vielzahl unterschiedlicher Problemszenarios.
- Eine generische automatische Behandlung der möglichen Probleme ist nicht möglich.
- Daher:
Das Auftreten der Fehler muss vermieden werden.

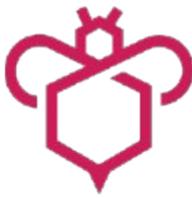


Vorher auf laufende Software prüfen:

- Produkt muss die Prozesse kennen:
 - Das Installationverzeichnis muß bekannt sein
 - Alle *.exe innerhalb des Installationsverzeichnis sind potentiell bedenkliche Prozesse
 - Optional können noch auf weitere Prozesse angegeben werden



- Wenn 'störender' Prozess läuft - mögliche **automatisierte** Reaktion:
 - Installation als gescheitert abbrechen (keine gute Idee)
 - Prozesse automatisiert beenden und installieren (auch keine gute Idee)
 - Vor Installation abbrechen und auf später hoffen (besser)



Wenn 'störender' Prozess läuft - mögliche **interaktive** Reaktion:

- Interaktiver Dialog (mit Timeout und Default)
 - Aufforderung zum manuellen Beenden des Prozesses (und danach Installation)
"Nochmal probieren" (recheck)
 - Installation verschieben und auf später hoffen
"Jetzt nicht" (defer)
 - Prozesse automatisiert beenden und installieren
"Prozess beenden" (kill)

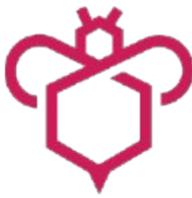


Problem

- Störung durch aufpopende Fenster



- Setupprogramme 'silent' aufrufen
- opsiclientd konfigurieren für Background Installation:
 - Event-Notifier nicht anzeigen
 - Andere Notifier bleiben sichtbar
 - opsi-script silent aufrufen



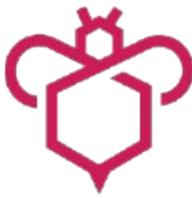
opsi-script 'silent' ausführen :

- Aktivität des opsi-script selbst ist nicht sichtbar
- Setup Meldungen bleiben sichtbar
- Besondere Meldungen des opsi-script können angezeigt werden



Umfangreiche Arbeiten :

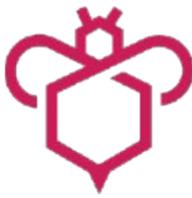
- opsiconfd Konfiguration
- Script Anpassungen (silent Aufrufe)



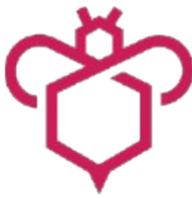
Keine Änderungen am Script und Konfiguration:

- opsi-script nur auf dem winlogon Desktop
- opsi-event Notifier nur auf dem winlogon Desktop
- Besondere Meldungen des opsi-script werden über den opsiclientd auf dem current Desktop angezeigt

Das ist die Variante welche umgesetzt wird !



- Anforderungen
- Stand



- Einbettung in eine Gesamtstrategie der Entwicklung:
Also nichts, was später als 'Altlast' stört.
- Komplexität verringern (nicht erhöhen).
- Nutzung erst sinnvoll möglich,
wenn alle verwendete Pakete das können
- Vorhandene Pakete müssen daher einfach für die
Background-Installation enabled werden können.



Darstellung des Standes der
experimentellen Entwicklung



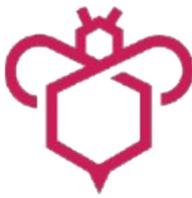
- Fehler vermeiden



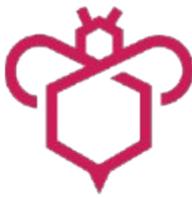
- Kritische Prozesse erkennen:
 - Installationsverzeichnis
 - sonstige Prozesse
- Nicht im Script
- Notwendige Daten sind ein Bestandteil der Metadaten des Produktes



- Metadaten des Produkts
 - `control.toml` / Backend
 - Nicht schnell umsetzbar
- Neue Datei mit 'zusätzlichen' Metadaten
- Im Produktverzeichnis (`CLIENT_DATA`)
- TOML Format
- In Zukunft: Übernahme der Metadaten in das Backend



- Metadaten nicht nur für Background Install
- Entwurf der Struktur dieser Metadaten ist damit auch Bestandteil des Entwurfs der zukünftigen Produkt-Datenstruktur
- Daher Gegenstand intensiver Überlegungen und wahrscheinlich auch Veränderungen



Verschieben von Installationen:

- Neuer Status (neben `success`, `failed`, `suspended`):
`'deferred'`
 - Status: unverändert (`'installed'`)
 - Bericht: `'deferred'`
 - Actionrequest: bleibt erhalten (`'setup'`)
 - Version: unverändert



- Wenn kritischer Prozeß läuft,
so wird ein Dialog mit einem Timeout angezeigt.
Timeout = 60 Sekunden
- Auswahloptionen:
 - `recheck` - "Nochmal probieren"
 - `defer` - "Jetzt nicht"
 - `kill` - "Prozess beenden"
- `Defer` ist der Default



Implementierung im opsi-script

- Lesen der Metadaten vor der Scriptausführung
- Prüfung auf kritische Prozesse vor (!) (De-)Installation
- Zu Testzwecken aus Bibliothek:
"uib_win_background_install.opsilib"

```
importlib "uib_win_background_install.opsilib"
```

- Final: Direkt im opsi-script implementiert

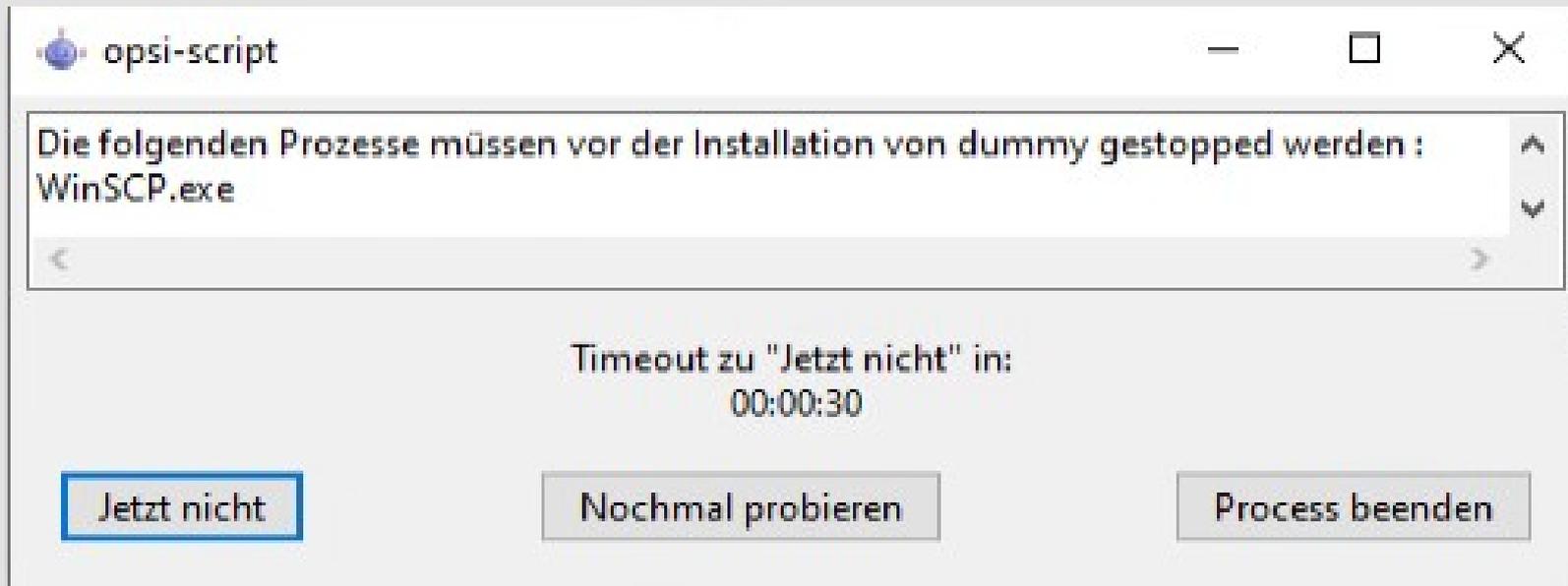


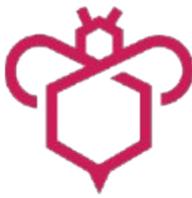
Implementierung im opsi-script

- Lesen der Metadaten aus TOML-Datei (initiale Implementierung)
- Ermitteln der möglichen Prozesse aus dem Installationsverzeichnis
- Funktion für Prozessliste mit Pfaden
- Setzen des Status deferred
- Interaktiver Dialog (provisorisch direkt vom opsi-script)



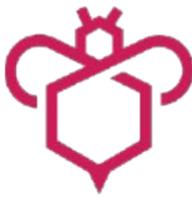
- Interaktiver Dialog (experimentell, aus opsi-script):





Konfigurierbare Defaults über configs:

- Default nach Timeout ist `defer` ('Jetzt nicht')
(`TIMEOUT_ACTION=defer`)
- Timeout ist 60 Sekunden
(`DIALOG_TIMEOUT=60`)



Konfigurierbare Defaults über configs:

- Nach 3 mal recheck in Folge (`MAXRECHECK=3`),
wird der Prozeß deferred (`MAXRECHECK_ACTION=defer`)
- Nach 5 mal deferred in Folge (`MAXDEFER=5`),
wird der Prozeß gekillt und mit der Installation
fortgefahren (`MAXDEFER_ACTION=kill`)
Ist das schlau ?



- Störung durch aufpopende Fenster



Keine Änderungen am Script und Konfiguration nötig

- opsi-script nur auf dem winlogon Desktop ✓
- opsi-event Notifier nur auf dem winlogon Desktop ✓
- Interaktive Dialoge des opsi-script werden über den opsiclientd auf dem current Desktop angezeigt ✗



Vorführung



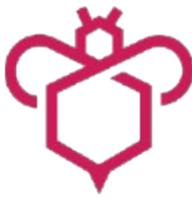
Offene Fragen



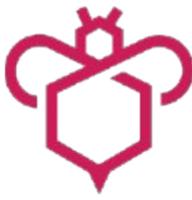
- Metadaten:
Installdirs von alten Versionen
der zur installierenden Software (soweit abweichend)
- Metadaten:
Installdirs von alten Versionen via regex aus Registry:
Uninstall Einträge der Software anhand des
'DisplayName' finden und dann
die 'InstallLocation' einsammeln.



- Prüfung auf störende Prozesse immer durchführen ?
Auch vor dem Login ?
- Wie umgehen mit Produkten welche nicht Background enabled sind ?
- Erkennen von Background Situationen und
Beschränkung auf Background enabled Produkte ?



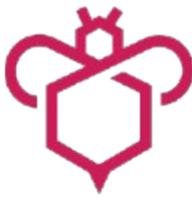
- Prozesse aus laufenden Diensten werden nicht angezeigt.
Es wird davon ausgegangen das der Installer damit umgehen kann.
Korrekt ?
- Probleme bei WAN Erweiterung ?
- Gibt es ein Problem mit evtl. nicht aufgelösten Abhängigkeiten, weil abhängiges Produkt deferred wurde ?
- Brauchen wir sowas für Linux, Mac ?



- Brauchen wir eine produktbezogene Konfiguration für:
 - Default Aktion (`TIMEOUT_ACTION`)
 - `MAXRECHECK`, `MAXDEFER`, `DIALOG_TIMEOUT`
 - `MAXRECHECK_ACTION`, `MAXDEFER_ACTION`
 - Wenn ja - über Metadaten oder über Produktproperty ?
 - Was wäre dann die Konfigurationshierarchie: (default < config < produkt) ?



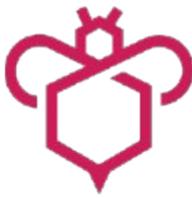
- Soll die Interaktivität abschaltbar sein ?
`DIALOG_TIMEOUT = 0`
- Brauchen wir weitere mögliche Actions ?
z.B.: `fail`
- Soll die Aktivität trotz verstecken noch sichtbar sein ?
 - Notification ?
 - Systray ?



- Wie funktioniert das Konzept bei Terminalservern
Soll es hier überhaupt anwendbar sein ?
- Workstations mit mehreren Usern (teilweise inaktiv)
Prozessliste mit user Kontext
- Soll es überhaupt eine Interaktivität geben,
wenn mehrere Instanzen eines Prozesses mit
unterschiedlichen Ownern laufen ?



- Werkzeuge zur Erstellung der Metadaten Datei:
- Grafisches Tool:
 - Button zur Auswahl des Installdir
 - Editfeld zur Anzeige bzw. Eingabe des Installdir
 - Button zum Speichern der Metadatei
- Kommandozeilen Werkzeug ?



Roadmap: Preview in opsi 4.4



Danke für die Aufmerksamkeit