



Vivaldi

ToC

- [Paketinfo](#)
 - [Paket erstellen](#)
 - [Voraussetzungen](#)
 - [Makefile und spec.json](#)
 - [Mustache](#)
 - [Verzeichnisstruktur](#)
 - [Makefile-Parameter](#)
 - [spec.json](#)
 - [Paketerstellung](#)
 - [Installation](#)
 - [Allgemeines](#)
 - [Aufbau des Paketes](#)
 - [Nomenklatur](#)
 - [Unattended-Switches](#)
 - [Lizenzen](#)
 - [Dieses Paket](#)
 - [psdetail](#)
 - [7zip](#)
 - [Vivaldi](#)
 - [Anmerkungen/ToDo](#)
-

Paketinfo

Das vorliegende OPSI-Paket fuer **Vivaldi** wurde fuer das Repository von *opsi4institutes (O4I)* entwickelt.

Die Erstellung der eigentlichen OPSI-Pakete aus den Quellen erfolgt durch ein einfaches *Makefile*.

Paket erstellen

Dieser Abschnitt beschaeftigt sich mit der Erstellung des OPSI-Paketes aus dem Source-Paket und nicht mit dem OPSI-Paket selbst.

Voraussetzungen

Zur Erstellung der OPSI-Pakete aus den vorliegenden Quellen werden neben den **opsi-utils** noch weitere Tools benoetigt, die aus den Repositories der jeweiligen Distributionen zu beziehen sind. Das sind (angegebenen Namen entsprechen Paketen in Debian/Ubuntu):

- make
- curl oder wget
- mustache (im Repository enthalten)

Makefile und spec.json

Da aus den Quellen verschiedene Versionen des Paketes mit entsprechenden Anpassungen generiert werden sollen (intern, O4I; testing/release) wurde hierfuer ein **Makefile** erstellt. Darueber hinaus steuert **spec.json** die Erstellung der Pakete.

Im Idealfall ist beim Erscheinen einer neuen Release von Vivaldi lediglich die **spec.json** anzupassen.

Mustache

Als Template-Engine kommt **Mustache** zum Einsatz.

Im Detail wird hier eine Go-Implementierung verwendet. Die Software ist auf [Github](#) zu finden. Binaries für Linux und Windows liegen diesem Paket bei.

Das in vorherigen Versionen dieses Paketes (<11) verwendete `pystache` kommt nicht mehr zum Einsatz und wurde aus den Quellen entfernt.

Verzeichnisstruktur

- **PACKAGES** - erstellte Pakete
- **DOWNLOAD** - heruntergeladene Installationsarchive und md5sums
- **BUILD** - Arbeitsverzeichnis zur Erstellung der jeweiligen Pakete
- **SRC** - Skripte und Templates

Einige Files werden bei der Erstellung erst aus `.in`-Files generiert, welche sich in den Verzeichnissen `SRC/OPSI` und `SRC/CLIENT_DATA` befinden. Die `SRC`-Verzeichnisse sind in den `OPSI`-Paketen nicht mehr enthalten.

Makefile-Parameter

Der vorliegende Code erlaubt die Erstellung von `OPSI`-Paketen fuer die Releases gemaess der Angaben in `spec.json`. Es kann jedoch bei der Paketerstellung ein alternatives `Spec`-File uebergeben werden:

`SPEC=<spec_file>`

Das Paket kann mit `"batteries included"` erstellt werden. In dem Fall erfolgt der Download der Software beim Erstellen des `OPSI`-Paketes und nicht erst bei dessen Installation:

`ALLINC=[true|false]`

Standard ist hier die Erstellung des leichtgewichtigen Paketes (`ALLINC=false`).

Bei der Installation des Paketes im Depot wird ein eventuell vorhandenes `files`-Verzeichnis zu-naechst gesichert und vom `postinst`-Skript spaeter wiederhergestellt. Diese Verzeichnis beinhaltet die eigentlichen Installationsfiles. Sollen alte Version aufgehoben werden, kann das ueber einen Parameter beeinflusst werden:

`KEEPFILES=[true|false]`

Standardmaessig sollen die Files geloescht werden.

`OPSI` erlaubt des Pakete im Format `cpio` und `tar` zu erstellen.

Als Standard ist `cpio` festgelegt.

Das `Makefile` erlaubt die Wahl des Formates ueber die Umgebungsvariable bzw. den Parameter:

`ARCHIVE_FORMAT=<cpio|tar>`

spec.json

Haeufig beschraenkt sich die Aktualisierung eines Paketes auf das Aendern der Versionsnummern und des Datums etc. In einigen Faellen ist jedoch auch das Anpassen weiterer Variablen erforderlich, die sich auf verschiedene Files verteilen.

Auch das soll durch das `Makefile` vereinfacht werden. Die relevanten Variablen sollen nur noch in `spec.json` angepasst werden. Den Rest uebernimmt `make`

Paketerstellung

Soll lediglich die Software aktualisiert werden, beschränken sich die notwendigen Änderungen in der `spec.json` auf:

- `O_SOFTWARE_VER`
- `O_CHANGELOG`

Weiterhin sollte die changelog um einen entsprechenden Eintrag ergänzt werden.

Gültige Targets für die Paketerstellung liefert

```
make help
```

In der Regel ist *all_prod* die passende Wahl:

```
make all_prod
```

erstellt.

Sofern erforderlich erfolgt nun der Download der Software von den Vivaldi-Servern automatisch.

Das Target *all_prod* erstellt 3 Pakete: * *vivaldi* (hausinternes Download-Paket des MPIMSP) * *dfn_vivaldi* (leichtgewichtiges *self-download*-Paket für das o4i-Repository) * *o4i_vivaldi* (*batteries-included*-Paket für das o4i-Repository)

Installation

Das Makefile erlaubt die Erstellung verschiedener Varianten des OPSI-Paketes. Details hierzu liefert **make help**

Standardmässig werden Pakete erstellt, bei denen die Software selbst nicht mit dem OPSI-Paket vertrieben wird. Dennoch ist ein manueller Download der Software hier nicht erforderlich. Bei der Installation des Paketes im Depot erfolgt im *postinst*-Script der Download der Software vom Hersteller (Windows, 32 und 64 Bit). Entsprechende Pakete sollten i.d.R. durch einen Suffix “~dl” gekennzeichnet sein (siehe [Nomenklatur](#)). Beim Download der erforderlichen Files erfolgt eine Ueberpruefung der MD5-Summen; diese sind im OPSI-Paket hinterlegt. Alternativ lassen sich auch “*batteries included*”-Pakete per Makefile erstellen. In diesen ist - wie zu vermuten - die Software selbst bereits enthalten.

“*batteries included*”-Pakete und “*self download*”-Pakete sind nach der Installation im Depot technisch identisch.

Die Aktivitaeten von *preinst* und *postinst* werden auf dem Depot-Server in einem Logfile protokolliert. Standardmaessig (definiert in *spec.json*) ist dieses unter ***/tmp/\${PRODUCT_ID}__opsi_package_install.log*** zu finden.

Hier eventuell auftretende Fehler werden an den *opsi-package-manager* uebergeben und setzen das Paket in einen Fehlerzustand.

Mit

```
make install
```

können alle zuvor für die aktuelle Version erstellten Pakete auf dem Depot-Server installiert werden.

Allgemeines

Aufbau des Paketes

- **variables.opsiinc** - Da Variablen ueber die Scripte hinweg mehrfach verwendet werden, werden diese (bis auf wenige Ausnahmen) zusammengefasst hier deklariert.
- **product_variables.opsiinc** - die produktspezifischen Variablen werden hier definiert
- **setup.opsiscript** - Das Script fuer die Installation.
- **uninstall.opsiscript** - Das Uninstall-Script
- **delsub.opsiinc** - Wird von Setup und Uninstall gemeinsam verwendet. Vor jeder Installation/jedem Update wird eine alte Version entfernt. (Ein explizites Update-Script existiert derzeit nicht.)
- **checkinstance.opsiinc** - Pruefung, ob eine Instanz der Software laeuft. Gegebenenfalls wird das Setup abgebrochen. Optional kann eine laufende Instanz zwangsweise beendet werden.
- **checkvars.sh** - Hilfsscript fuer die Entwicklung zur Ueberpruefung, ob alle verwendeten Variablen deklariert sind bzw. nicht verwendete Variablen aufzuspüren.
- **bin/** - Hilfsprogramme; hier: **7zip**, **psdetail**
- **images/** - Programmbilder fuer OPSI

Nomenklatur

Praefixes in der Produkt-Id definieren die Art des Paketes:

- **0_** oder **test_** - Es handelt sich um ein Test-Paket. Beim Uebergang zur Produktions-Release wird der Praefix entfernt.
- **o4i_** oder **dfn_** - Das Paket ist zur Verwendung im O4I/DFN-Repository vorgesehen.

Suffix:

- **~dl** - Das Paket enthaelt die Installationsarchive selbst nicht. Diese werden erst bei der Installation im Depot vom postinst-Skript heruntergeladen.
Der Suffix ist kein Bestandteil der ProductId. Nach der Installation im Depot gibt es hierauf keinerlei Hinweis mehr.

Die Reihenfolge der Praefixes ist relevant; die Markierung als Testpaket ist stets fuehrend.

Unattended-Switches

...gibt es derzeit beim Vivaldi-Installer nicht. Die Installation besteht daher aus dem manuellen entpacken des Archivs und Erzeugen der Startmenueeintraege.

In der Folge gibt es auch keinen Eintrag in den Uninstall-Sektionen der Registry.

Damit OPSI sich dennoch "orientieren" und den Installationsstatus abfragen kann, wird unter *PackageReg* ein Eintrag fuer derart installierte Pakete angelegt.

Lizenzen

Dieses Paket

Dieses OPSI-Paket steht unter der *GNU General Public License* **GPLv3**.

Ausgenommen von dieser Lizenz sind die unter **bin/** zu findenden Hilfsprogramme. Diese unterliegen ihren jeweiligen Lizenzen.

psdetail

Autor der Software: Jens Boettge <boettge@mpi-halle.mpg.de>

Die Software **psdetail.exe** wird als Freeware kostenlos angeboten und darf fuer nichtkommerzielle sowie kommerzielle Zwecke genutzt werden. Die Software darf nicht veraendert werden; es duerfen keine abgeleiteten Versionen daraus erstellt werden.

Es ist erlaubt Kopien der Software herzustellen und weiterzugeben, solange Vervielfaeltigung und Weitergabe nicht auf Gewinnerwirtschaftung oder Spendensammlung abzielt.

Haftungsausschluss:

Der Auto lehnt ausdruecklich jede Haftung fuer eventuell durch die Nutzung der Software entstandene Schaeden ab.

Es werden keine ex- oder impliziten Zusagen gemacht oder Garantien bezueglich der Eigenschaften, des Funktionsumfanges oder Fehlerfreiheit gegeben.

Alle Risiken des Softwareeinsatzes liegen beim Nutzer.

Der Autor behaelt sich eine Anpassung bzw. weitere Ausformulierung der Lizenzbedingungen vor.

Fuer die Nutzung wird das *.NET Framework ab v3.5* benoetigt.

7zip

Es gilt die Lizenz von <http://www.7-zip.org/license.txt>.

Die Lizenz liegt diesem Paket in CLIENT_DATA/bin/ ebenfalls bei.

Vivaldi

Vivaldi steht unter [BSD-Lizenz](#) mit proprietären Bestandteilen.

Das verwendete Vivialdi-Logo ist gemeinfrei.

Quelle: [https://de.wikipedia.org/wiki/Vivaldi_\(Browser\)#/media/File:Vivaldi_web_browser_logo.svg](https://de.wikipedia.org/wiki/Vivaldi_(Browser)#/media/File:Vivaldi_web_browser_logo.svg)

Anmerkungen/ToDo

- Die vollstaendige Integration des Browsers ins System (Default-Browser, HTML-Handler, Self-Updater erfolgt aufgrund der Art der Installation derzeit nicht).
- Die Product-Property *default_language* wird derzeit nicht ausgewertet uns ist daher deaktiviert
- Policies fuer Chromium/Vivaldi sind bislang noch nicht realisiert.

Jens Boettge <boettge@mpi-halle.mpg.de>, 2023-09-13 09:26:30 +0200