



Double Commander

ToC

- [Paketinfo](#)
- [Paket erstellen](#)
 - [Makefile und spec.json](#)
 - [Quickstart](#)
 - [pystache](#)
 - [Verzeichnisstruktur](#)
 - [Makefile-Parameter](#)
 - [spec.json](#)
- [Installation](#)
- [Properties](#)
- [Allgemeines](#)
 - [Aufbau des Paketes](#)
 - [Nomenklatur](#)
 - [Unattended-Switches](#)
- [Lizenzen](#)
 - [Dieses Paket](#)
 - [Double Commander](#)
 - [psDetail](#)
 - [GetRealName](#)
 - [7Zip](#)
- [Anmerkungen/ToDo](#)

Dieses OPSI-Paket fuer **Double Commander** wurde fuer das *O4I*-Repository und das Repositorys des *Max-Planck-Instituts fuer Mikrostrukturphysik* erstellt.

Es wird versucht auf die Besonderheiten der jeweiligen Repositories einzugehen; entsprechend werden durch ein einfaches *Makefile* aus den Quellen verschiedene Pakete erstellt.

Paket erstellen

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

Makefile und spec.json

Da aus den Quellen verschiedene Versionen des Paketes mit entsprechenden Anpassungen generiert werden sollen (mpimsp/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 des *Double Commander* lediglich die **spec.json** anzupassen.

Quickstart

Sind die Voraussetzungen erfuehlt, d.h. das *.spec-File* wurde an die aktuelle Version angepasst und *pystache* ist vorhanden, kann das Paket mit

```
make o4i install
```

erstellt und im Anschluss gleich auf dem Depot-Server installiert werden.
Statt *o4i* kann auch *mpimsp* oder *all_prod* zum Einsatz kommen.
Details zu weiteren Targets verrät **make help**.

pystache

Als Template-Engine kommt **pystache** zum Einsatz. Das entsprechende Paket ist auf dem Build-System aus dem Repository der verwendeten Distribution zu installieren.

Unter Debian/Ubuntu erledigt das:

```
sudo apt-get install python3-pystache
```

Verzeichnisstruktur

Die erstellten Pakete werden im Unterverzeichnis **BUILD** abgelegt.

Einige Files (derzeit *control*, *preinst*, *postinst*) 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 gemäss 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 auf dem Depot-Server:

```
ALLINC={true, false}
```

Standard ist hier die Erstellung des Download-Paketes (*ALLINC=true*).

Bei der Installation des Paket im Depot wird ein eventuell vorhandenes *files*-Verzeichnis zunaechst 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 vor Version 4.3 erlaubte es Pakete im Format *cpio* oder *tar* zu erstellen. Mit Version 4.3 steht nur noch *tar* zur Verfuegung, weshalb dieser Parameter eigentlich obsolet ist.

Als Standard fuer dieses Paket ist nun *tar* festgelegt.

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

```
ARCHIVE_FORMAT={cpio, tar} (1)
```

⁽¹⁾ Fuer *OPSI* 4.3 wird durch das Makefile ebenfalls nur noch *tar* unterstuetzt; fuer 4.2 steht *cpio* noch zur Verfuegung.

Als Kompressionsformat koennen *gz*, *gzip*, *bz2*, *bzip2*, oder *zstd* zum Einsatz kommen.

Fuer Kompatibilitaet zu *OPSI* <4.3 sollte vorerst *gzip* verwendet werden. (Mit *zstd* gab es teilweise Probleme.)

Als Standard fuer dieses Paket ist *gz* festgelegt.

OPSI 4.3 verwendet normalerweise `zstd`.

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

```
COMPRESSION={gzip,zstd}  
COMPRESSION={gz,gzip,zstd,bz2,bzip2} (ab OPSI 4.3)
```

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`

Installation

Die Software selbst wird in der **~dl**-Version (`ALLINC=false` bei `make`) nicht mit diesem Paket vertrieben. Fuer die standardmaessig erstellten *"batteries included"*-Pakete entfaellt dieser Abschnitt.

Je nach Art des erstellten Paketes erfolgt bei der Installation im Depot durch das `postinst`-Script der Download der Software vom Hersteller (Windows, 32 und 64 Bit).

Ein manueller Download sollte dann nicht erforderlich sein.

Auf dem Depot-Server ist **curl** bzw. **wget** erforderlich.

Das Gesamtvolumen der herunterzuladenden Dateien betraegt ca. **20 MByte**.

Es werden in diesem Paket die **INNO**- anstelle der MSI-Versionen der Installer verwendet, da letztere bislang keine Anpassung des Zielverzeichnis erlauben.

Properties

Die Installation der Software laesst sich ueber eine Reihe von Properties beeinflussen:

- **custom_default_config** - Hier kann der Dateiname fuer eine Default-Konfiguration angegeben werden. Diese muss im Verzeichnis `config` oder `custom` liegen und ueberschreibt die vom Installer angelegte Minimalkonfiguration (`doublecmd.xml`).
Konfigurationen im `custom`-Verzeichnis werden bei Namensgleichheit bevorzugt behandelt und bei Aktualisierung des Paketes nicht ueberschrieben.
- **custom_post_install** - Das hier angegebene OPSI-Script wird ins Setup eingebunden und im Anschluss an die Installation ausgefuehrt.
- **custom_post_uninstall** - Das hier angegebene OPSI-Script wird in die Deinstallation eingebunden und im Anschluss ausgefuehrt.
- **default_language** - Hier kann fuer die Applikation eine Voreinstellung fuer die Sprache gewaehlt werden. Diese kann vom Anwender spaeter individuell angepasst werden.
In der Voreinstellung wird versucht die Systemsprache zu verwenden. Darueber hinaus kann zwischen *de*, *en*, *fr*, *it*, *es* gewaehlt werden.
- **install_architecture** - Hier kann zwischen der 32- und der 64-Bit-Version der Software gewaehlt werden. Standardmaessig wird die zum Betriebssystem passende Version gewaehlt.
- **kill_running** - Erfolgt die (De)Installation *on demand* und nicht beim Boot/Shutdown, ist es moeglich, dass eine Instanz der Software laeuft.

Hiermit ist es moeglich diese Zwangsweise zu beenden. Andernfalls schlaegt die Installation fehl.

- **link_desktop** - Ermoeoglicht das Anlegen einer Desktopverknuepfung.
- **log_level** - Setzt einen alternativen Loglevel fuer das Paket. Fuer *Releases* ist der Standardwert 5, fuer *Testpakete* 7.

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**, **GetRealName**.
 - **images/** - Programmbilder fuer OPSI
 - **config/** - Default-Konfigurationen zur Auswahl und Anpassung
 - **custom/** - optionale Konfigurationen und Custom-Scripte.
- Die Daten dieses Verzeichnisses bleiben bei einer Paketaktualisierung erhalten.

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. (Achtung: Es auch moeglich Test-Pakete ohne expliziten Praefix zu erstellen.)
- **o4i_** - Das Paket ist zur Verwendung im Opsi4Institutes-Repository vorgesehen.
- **dfn_** - Das Paket ist zur Verwendung im Opsi4Institutes-Repository vorgesehen. (identisch mit o4i, Legacy-Paket!)

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

Unattended-Switches

Es handelt sich um ein *INNO*-Setup-Paket mit den hier gebrauchlichen Parametern. siehe auch: <http://www.jrsoftware.org/isinfo.php>

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.

Double Commander

Autor der verwendeten Logos ist Andryei Gudyak. Als Bestandteil der Software steht es unter der *GNU General Public License* **GPLv2**. Die Variationen fuer das OPSI-Paket wurden von mir unter Verwendung weiterer freier Grafiken erstellt

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.

GetRealName

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

Die Software **GetRealName.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.

7zip

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

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

Anmerkungen/ToDo

siehe [Git-Issues](#)

Jens Boettge <boettge@mpi-halle.mpg.de>, 2024-04-02 08:20:49 +0200