



Amazon Corretto 8 / 11 / 15 / 16 / 17

ToC

- [Paketinfo](#)
 - [Amazon Corretto](#)
- [Paket erstellen](#)
 - [Voraussetzungen](#)
 - [Makefile und spec.json](#)
 - [Mustache](#)
 - [Verzeichnisstruktur](#)
 - [Makefile-Parameter](#)
 - [spec.json](#)
- [Installation](#)
 - [Abhaengigkeiten](#)
- [Allgemeines](#)
 - [Properties](#)
 - [Aufbau des Paketes](#)
 - [Nomenklatur](#)
- [Lizenzen](#)
 - [Dieses Paket](#)
 - [Amazon-Corretto-Lizenzen](#)
 - [psDetail](#)
 - [GetRealName](#)
 - [7zip](#)
 - [Logo](#)
- [Anmerkungen/ToDo](#)

Paketinfo

Dieses OPSI-Paket (bzw. dessen Quellen) fuer Java-LTS-releases **Amazon Corretto 8**, **Amazon Corretto 11** und **Amazon Corretto 17** sowie fuer die nicht-LTS-Releases **Amazon Corretto 15** und **Amazon Corretto 16** deckt das *Java Development Kit* inklusive *Runtime* fuer die jeweiligen Releases ab. Eine separate JRE wird nicht mehr angeboten.

Waehrend Java 8 in einer 32- und 64-Bit-Version verfuegbar ist, liegt ab Versionen 11 nur noch eine Version fuer 64 Bit vor.

Teile dieser Dokumentation beziehen sich nicht ausschliesslich auf die erstellten OPSI-Pakete, sondern beruecksichtigen auch den Build-Prozess.

Amazon Corretto

Amazon Corretto ist eine kostenlose, plattformübergreifende und produktionsbereite Distribution des Open Java Development Kit (OpenJDK). Corretto wird von langfristigem Support begleitet, der

Leistungs- und Sicherheitsverbesserungen umfasst wird. Corretto ist mit dem Java SE-Standard kompatibel zertifiziert und wird intern bei Amazon für viele Produktionsservices verwendet.

Paket erstellen

Dieser Abschnitt beschäftigt 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 benötigt, die aus den Repositories der jeweiligen Distributionen zu beziehen sind. Das sind (angegebenen Namen entsprechen Paketen in Debian/Ubuntu):

- make
- curl bzw. wget

Makefile und spec.json

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

Im Idealfall sind beim Erscheinen einer neuen Release lediglich die jeweiligen json-Files anzupassen.

Ohne explizite Angabe eines json-Files wird **spec.json** verwendet. Dies erstellt generische Pakete ohne Angabe der Major-Release in der ProductId. Die spec_corretto*.json resultieren in Paketen, in denen die Major-Release Bestandteil der ProductId ist. Dies ermöglicht die parallele Installation mehrerer Releases.

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

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

Einige Files (control, postinst, setup.opsiscript) 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 für die Corretto-Releases **8**, **11**, **16** und **17**. Die Auswahl erfolgt über das entsprechende SPEC-File. Mitgeliefert werden `""spec.json""` (Corretto11, ProductId amazon-corretto) sowie `""spec_corretto[8,11,16,17].json""` (ProductIds jeweils mit Major-Release):

`SPEC=<spec_file>`

Ohne Angabe des Parameters werden die Pakete für Corretto 11 erstellt.

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`). Zuvor sollten jedoch die Installationspakete mit **make download** (ggf. unter Angabe eines Spec-Files) heruntergeladen werden, da diese fuer die Berechnung der Pruefsummen benoetigt werden.

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`

Installation

Die Software selbst wird - sofern bei der Paketerstellung nicht anders vorgegeben - nicht mit diesem Paket vertrieben. Fuer die *“batteries included”*-Pakete entfaellt dieser Abschnitt.

Bei der Installation des Paketes im Depot erfolgt im `postinst`-Script der Download der Software vom Hersteller (Windows, 32 (sofern vorhanden) 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 je nach Paketvariante zwischen **170** und **220 MByte**.

Da die Pakete von *lokalen Funktionen* Gebrauch machen, wird auf dem Depot-Server *opsi-winst* mindestens in der Version 4.12(.0.13) vorausgesetzt.

Abhaengigkeiten

Fuer weitere Hilfsprogramme ([psDetail](#) und [GetRealName](#)) das **.NET-Framework** ab der Version 3.5 benoetigt.

Allgemeines

Properties

Je nach Art des erstellten Paketes und den Einstellungen in der `spec.json` koennen die verfuegbaren Properties abweichen.

Property	Type	Values	Default	Multivalue	Editable	Description	Note
custom_post_install	unicode	"none", "custom_test.opsiinc", "post-install.opsiinc"	"none"	False	True	Define filename for include script in custom directory after installation	
custom_post_uninstall	unicode	"none", "custom_test.opsiinc", "post-uninstall.opsiinc"	"none"	False	True	Define filename for include script in custom directory after deinstallation	
install_architecture	unicode	"32 bit", "64 bit", "sysnative"	"sysnative"	False	False	which architecture (32/64 bit) should be installed	only available for Corretto 8
kill_applic	unicode			True	True	Instead of killing only applications of this package, kill also these running applications; requires "kill_running; use suffix '.exe' or '%'" as wildcard	available if activated in spec.json
kill_running	bool		False			kill running instance (for software on_demand)	available if activated in spec.json
local_installer_copy	bool		False			Use a temporary local copy of the installation package to avoid delays through the network.	increases the required disk space

Property	Type	Values	Default	Multivalue	Editable	Description	Note
log_level	unicode	"default", "1", "2", "3", "4", "5", "6", "7", "8", "9"	"default"	False	False	Loglevel for this package	
set_env_java_home	boolean		True			Set Environment JAVA_HOME and PATH to InstallDir of jdk or jre?	
set_jar_handler	boolean		True			Set Corretto to default handler for JAR files? (unset on removal)	
silent_option	unicode	"silent", "very silent"	"silent"	False	False	Show (silent) or hide (very silent) progressbar of (un)installer	
uninstall_before_setup	boolean		True			Run uninstall before (re)installation	

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
- **helpers.opsifunc** - Bibliothek mit lokalen (Hilfs-)Funktionen.
- **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 aufzuspuren.
- **bin/** - Hilfprogramme; 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 DFN-Repository vorgesehen.

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

Suffix:

- ~dl - Das Paket enthaelt die Installationsarchive selbst nicht. Diese werden erst bei der Installation im Depot vom postinst-Skript heruntergeladen.

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.

Amazon-Corretto-Lizenzen

Corretto is released under the same open source license as OpenJDK, which is licensed under the GNU Public License version 2 with the Class Path Exception (GPLv2 with CPE). You can use Corretto as you would use OpenJDK.

Quelle: [Amazon Corretto FAQs](#)

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 Autor 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 Autor 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.

7zip

Es gilt die Lizenz von <http://www.7-zip.org/license.txt>. Die Lizenz liegt diesem Paket in CLIENT_DATA/bin/ ebenfalls bei.

Logo

Anregung fuer das erstellte Logo war:

<https://pixabay.com/de/java-pokal-kaffee-programmierung-151343>.

Die Variationen des Icon-Satzes fuer das OPSI-Paket wurden von mir unter Verwendung weiterer freier Grafiken erstellt.

Anmerkungen/ToDo

Jens Boettge <boettge@mpi-halle.mpg.de>, 2021-09-16 11:43:44 +0200