Konvertieren einer .jar-Datei in eine .exe-Datei mit Launch4j

Kathegorie
java, SQL Server

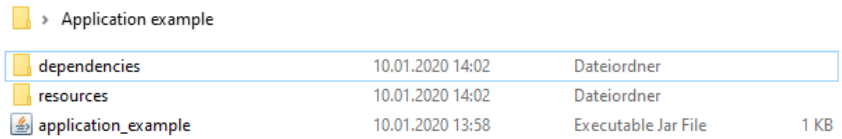
Aaron
Priesterroth

Um eine Java-Anwendung nativ in einer Windows Umgebung benutzen zu können, bietet es sich an, das lauffähige Programm mit Hilfe eines "Executable Wrappers" (eine "ausführbare Hülle") zu versehen, also eine ausführbare .exe-Datei zu erstellen.
Im Internet stehen zahlreiche Programme zur Verfügung, mit dessen Hilfe sich diese Konvertierung realisieren lässt. Eines davon ist Launch4j. Mit Hilfe von Launch4j kann in nur wenigen Schritten aus einer .jar- eine .exe-Datei erstellt werden.

Vorbereitung

Bevor Sie mit dem Übersetzen der .jar-Datei beginnen, sollten Sie die Strukturierung der Anwendung bezüglich der Quell-Dateien, Ressourcen und Abhängigkeiten bestimmen. Üblicherweise wird dabei der Ansatz verfolgt, dass sich die Anwendung selbst auf der höchsten Stufe der Hierarchie befindet und Ressourcen/Abhängigkeiten tiefer in die Hierarchie propagiert werden.
Eine sehr simple, beispielhafte Strukturierung könnte dabei wie folgt aussehen:

Es wird ein Ordner angelegt, der die Anwendung, die Ressourcen, die Abhängigkeiten und zu guter letzt den Executable Wrapper beinhalten soll.

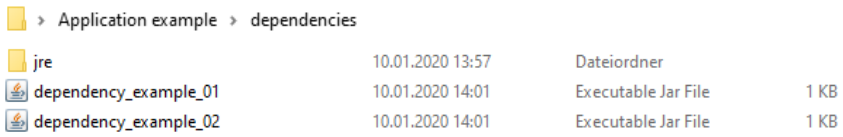


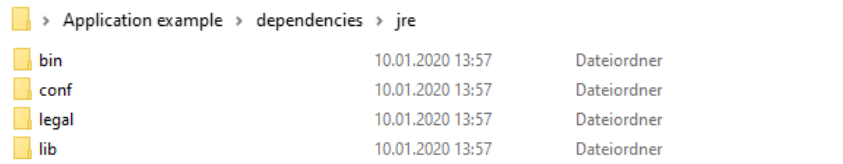
Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

In dem Ordner **resources** befinden sich die Ressourcen, die die Anwendung benötigt. Als Beispiel einer Java-Anwendung könnte es sich um fxml-Dateien, Bilder und/oder Datensätze, etc. handeln.

In dem Ordner dependencies befinden sich die Abhängigkeiten der Anwendung – etwa externe Bibliotheken, die von der Anwendung benötigt werden. Tiefer in der Hierarchie ausgehend von dependencies wird konventionell das benutzerdefinierte JRE (Java Runtime Environment) platziert, falls die Anwendung mit einem solchen ausgeliefert werden soll. Der Vorteil hierbei ist, dass der Benutzer keine Installation von Java benötigt um die Anwendung ausführen zu können.

Hinweis: In der Industrie übliche Namen für die Ordner resource und dependencies sind res und bin.





Hierbei ist es wichtig, darauf zu achten, dass die Hierarchie der Ressourcen mit den Vorgaben der Anwendung selbst übereinstimmen. Die Abhängigkeiten können beim Übersetzen mit Hilfe des sog. "Classpath" festgelegt werden.

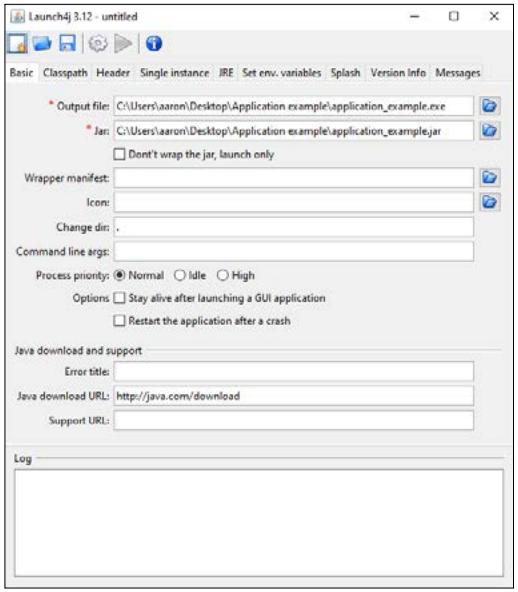
Konvertierung der .jar-Datei

Um mit dem Konvertieren der .jar-Datei zu beginnen, muss das Programm Launch4j gestartet werden.

Basic

In dem ersten Menüpunkt **Basic** müssen die folgenden Attribute spezifiziert werden:

- **Output file** – Der Ort, an dem die .exe-Datei abgelegt werden soll, sollte mit dem Ort der .jar-Datei übereinstimmen.
  Beispiel:
  Pfad der .jar-Datei: .../path/to/my/application_example.jar
  →
  Pfad der .exe-Datei: .../path/to/my/application_example.exe
- **Jar** – Der Ort, an dem sich die .jar-Datei befindet.
- **Change dir** – Der Einstiegspunkt der Anwendung im Dateisystem. Mit Rücksicht auf die Hierarchie sollte im Normallfall die Umgebung, in der sich die Anwendung befindet, festgelegt werden. Dies geschieht mit einem einzelnen Punkt ".".

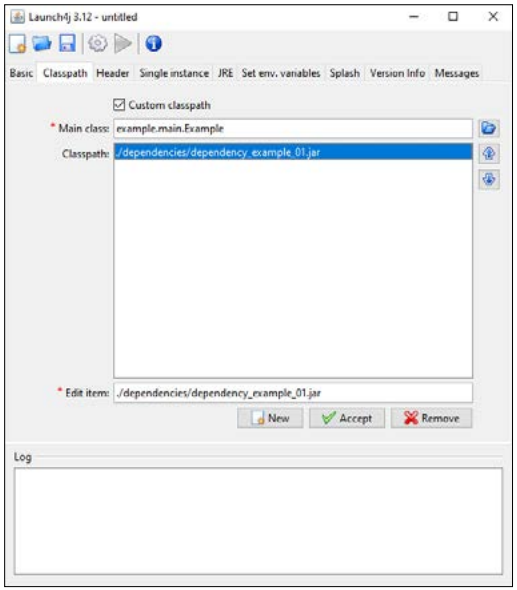


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Classpath

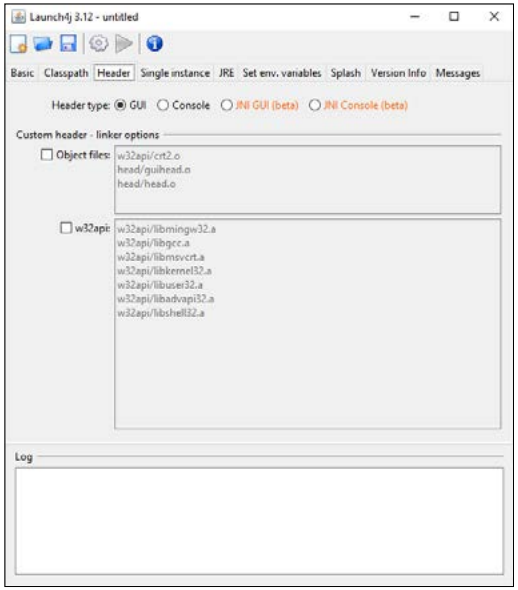
Der Menüpunkt **Classpath** wird dafür genutzt die Pfade der Abhängigkeiten anzugeben. Diese Funktion kann genutzt werden, wenn der eigentliche Classpath der Anwendung verändert wurde (wie beispielsweise in der oben beschriebenen Erstellung einer Hierarchie) und dieser angepasst werden muss.
Um einen benutzerdefinierten Classpath zu verwenden, muss das entsprechende Kontrollkästchen ausgewählt und die .jar-Datei im Feld **Main class** geöffnet werden.

Beispiel:
Der Pfad der Abhängigkeit muss angepasst werden, wenn die Java-Anwendung die Abhängigkeiten in unmittelbarer Nähe verlangt, diese sich jedoch in tieferer Hierarchie befinden. Somit muss der Pfad "dependency_example_01" in "**./dependencies/dependency_example_01**" umgeändert werden. Dies muss für jede Abhängigkeit geschehen.



Header

Im Menüpunkt **Header** kann die Art des Executable Wrappers ausgewählt werden. Hier gilt es hauptsächlich zwischen **GUI**, Graphical User Interface, eine Anwendung mit einer graphischen Oberfläche, oder **Console**, eine Anwendung, die über die Kommandozeile gesteuert wird, zu unterscheiden.



JRE

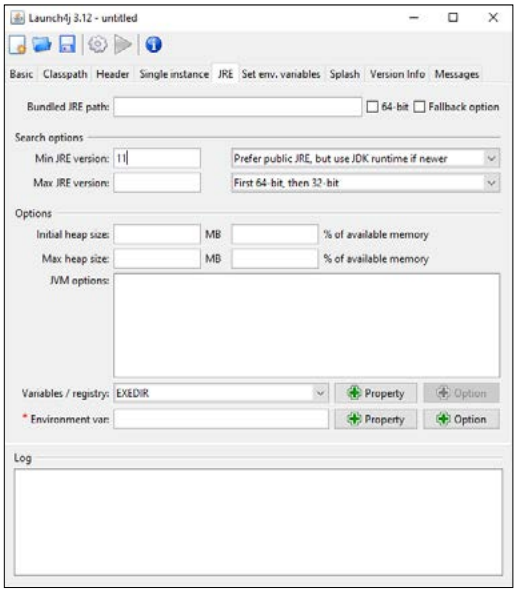
Im Menüpunkt **JRE** kann eine sog. bundled jre spezifiziert werden, die mit der Anwendung ausgeliefert wird. Dies ist optional.
Wird keine bundled jre angegeben, muss eine **minimum JRE version** angegeben werden. Dies hängt von der Java-Version ab mit der die Anwendung geschrieben wurde.

Besonders interessant ist an dieser Stelle das Feld für die Eingabe der **JVM Options**. An dieser Stelle können Argumente für die Java Virtual Machine angegeben werden.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Wrapper erzeugen

Mit einem Klick auf das Zahnrad im oberen Fensterrand kann der spezifizierte Wrapper erzeugt werden. Mit einem weiteren Klick auf das **Play-Symbol** rechts neben dem Zahnrad kann der erzeugte Wrapper auf Funktionalität getestet werden. Die bei dem Test erzeugten Nachrichten werden im Log-Fenster unten ausgegeben.



Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612