

Inhalt der von Alware gelieferten CD mit den Quellcodes für alles, was in JAVA programmiert ist, das Hilfsprogramm BAX (in C++) und für den UWM (in C++)

Eclipse	Enthält das Installationsprogramm für die Java.Entwicklungsumgebung
Java	Enthält Installationsprogramme für Java als Entwicklungsversion für Programmierer (SDK) und die Runtime-Environment (JRE) für Endbenutzer. Außerdem sind in gepackter Form die Quellcodes von Java für Programmierer beigelegt.
PLiG2	Das Lizenzierungsprogramm für Primero-Programme. Zum installieren Unter Windows.
Primero	Alle Quellcodes und relevanten Daten, die zur Weiterentwicklung von Primero nötig sind. Auch die Dokumentation der aktuellen Primero-Version (inkl. Primero-Licht) ist beigelegt: im Verzeichnis „doc“ die Datei index.html starten.
Primero-Licht	Alle Quellcodes und relevanten Daten, die zur Weiterentwicklung von Primero-Licht nötig sind.
Primero-Sommer-DIN	Alle Quellcodes und relevanten Daten, die zur Weiterentwicklung von Primero-Sommer-DIN nötig sind.
UWM	Alle Quellcodes und relevanten Daten, die zur Weiterentwicklung vom U-Wert-Manager nötig sind. Zur konkreten Weiterentwicklung wird jedoch noch das Microsoft Visual Studio .NET benötigt, welches nicht auf dieser CD ist.
BAX	Der Quellcode des Hilfsprogramms BAX.exe

Die weiterhin benötigten Quellcodes und Programme

DL_frac in Borland Delphi, von Detlef Hennings. Quellcode liegt an der HAW vor.

LichtdllJAVA.dll Ein sehr eckiges Stück! Zunächst gibt es den in FORTRAN geschriebenen Quellcode Lichtdll.for, von Udo Dietrich (liegt an der HAW vor). Daraus wird dann Mithilfe des FORTRAN Compilers und des C++ Compilers die dll erstellt, die das Lichtprogramm unter JAVA benötigt.

EnEV in FORTRAN, von Udo Dietrich. Quellcode liegt an der HAW vor. Einfach mit dem Fortran Compiler zur EnEV.exe compilieren.

Die Compiler

- Für Licht.exe und EnEV.exe
Fortran
Compaq Visual Fortran 6.6
Muss vorher installiert sein, nicht auf der CD von Alware.
- Für den UWM, das Erstellen der Lichtdll und BAX.exe
C++
MS Visual Studio.NET Version 2002
Muss vorher installiert sein, nicht auf der CD von Alware.
Achtung! Der Fortran Compiler und MS Visual Studio beeinflussen sich gegenseitig. Z.B. ist bei beiden Compilern die Dateieindung einer Projektarbeitsmappe .dsw. Beim Doppelklick auf eine solche Mappe wird treffsicher stets der zuletzt installierte Compiler gestartet. Im Moment ist dies MS Visual Studio, d.h. Fortran-Mappen kriegt man nur direkt über den Fortran Compiler auf.
- Die Entwicklungsumgebung für JAVA
Eclipse-SDK-3.0.1-win32.zip
Die auf der CD befindliche Datei (kann auch vom Netz gezogen werden) ist eine einfache .zip Datei, die bloß entpackt wird. Achtung! Es wird alles in ein Unterverzeichnis \eclipse abgelegt. Wenn man vorher bereits ein Verzeichnis \eclipse angelegt hat und von dort aus entpackt, haut er alles in \eclipse\eclipse....
Also kopieren in c:\ und dann entpacken!
Eclipse findet automatisch die installierte JAVA Version. Prüfen unter window/preferences/Java/Installed JREs
- Für JAVA
J2RE 1.4.2 ;
Auf der CD oder wird einfach vom Netz gezogen. Auf dem Notebook war diese JAVA-Version bereits mit WINDOWS XP drauf, Installation also nicht nötig.

Erfahrungen bei früheren Installationen:
Achtung! Bei der Installation von JAVA werden die notwendigen Pfadnamen standardmäßig NICHT angelegt. Entweder vereinbaren in System > Systemeigenschaften > Umgebungsvariablen > Systemvariablen und dann mit „neu“ anlegen „CLASSPATH“ und in die untere Zeile die Pfade für das Hauptverzeichnis, wo JAVA sitzt und die beiden Unterverzeichnisse \bin und \lib. Dieser von Thorsten vorgeschlagene Weg funktionierte nicht. Deshalb wurde JAVA einfach in das Verzeichnis installiert, in dem eclipse (verzweifelt) nach der JAVA-Datei sucht: C:\eclipse\jre\ - damit lief es dann!
- Für DL_frac.exe
Delphi
Borland Delphi Studio 7
Muss vorher installiert sein, nicht auf der CD von Alware.

Compilieren der JAVA – Quellcodes

1 PRIMERO-Licht

Voraussetzung: die Programme DL_frac.exe, BAX.exe und die LichtdllJAVA.dll sind bereits vorhanden (Beschreibung zu diesen Programmen weiter hinten!)

Falls notwendig, erst mal die richtige Ansicht in eclipse auswählen:
menü window >> open perspective >> java browsing

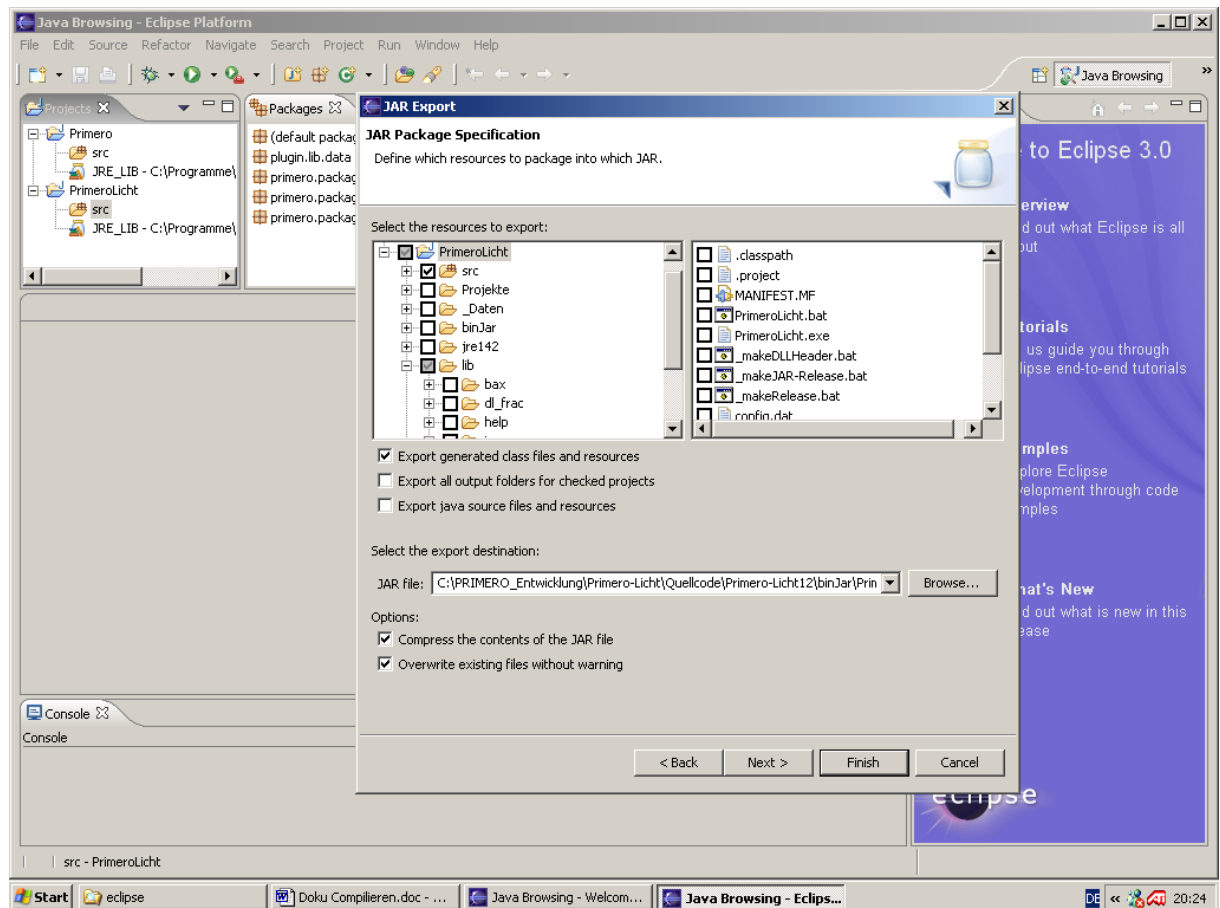
Das vorhandene Projekt in eclipse importieren
eclipse akzeptiert prinzipiell nur Projekte/Dateien/Quellcode, der über die Funktion importieren reingeholt wird.

dazu im Menü file >> import >> Existing Project into workspace next >> bei Project contents jetzt das Verzeichnis auswählen, in dem die Datei „.project“ liegt (hier „Primerio-Licht/Quellcode/Primerio-Licht12“) >> finish

Das Projekt wird dann geladen.

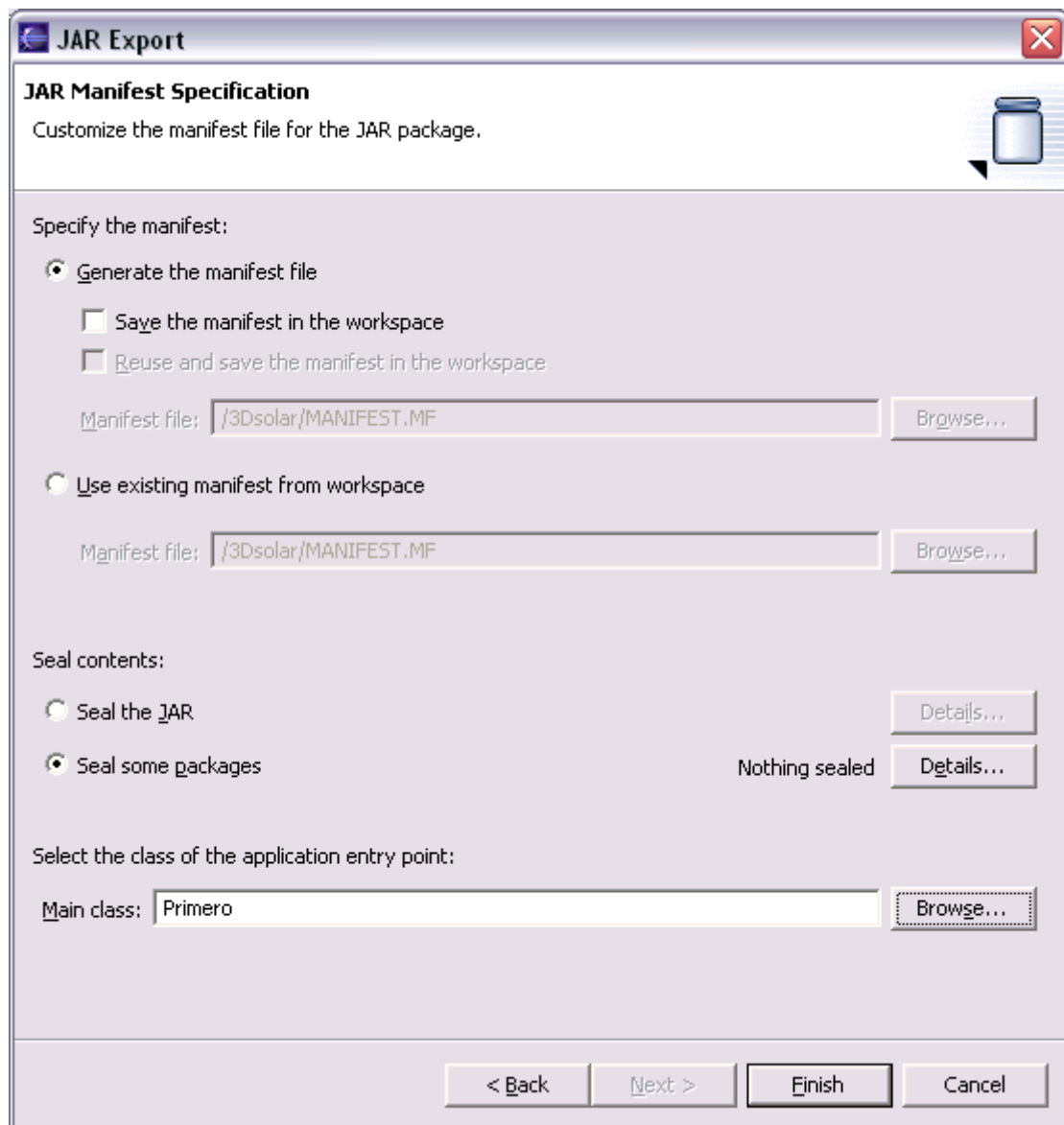
Erzeugen der Jar-Datei von Primerio-Licht

- Im Fenster **Projects** mit der rechten Maustaste auf den Baumeintrag **PrimerioLicht** klicken, dann im sich öffnenden Popup-Menü auf **Refresh** klicken
- Dasselbe danach mit dem Baumeintrag **src** vollziehen (befindet sich unterhalb des Eintrags **PrimerioLicht**)
- Mit der rechten Maustaste auf den Eintrag **src** im Baum klicken und im Menü auf **Export...**
- Im aufgehenden Fenster Doppelklick auf **JAR file**
- Nun den Baum für **Select the resources for export...** aufklappen (Pluszeichen neben Eintrag **PrimerioLicht**)
- Eintrag **src** ist schon markiert, so lassen, zusätzlich weiter unten noch ein Häkchen für Eintrag **lib/res** setzen
- Mit dem Windows-explorer das Verzeichnis Primerio-Licht/Quellcode/Primerio-Licht12/**binJar** anlegen (hier schon vorhanden)
- Nun noch den Pfad für die JAR-Datei angeben bei **JAR file:** **Wichtig** ist, dass dabei das (soeben angelegte) Verzeichnis binJar ausgewählt wird, in dem sich das Projekt befindet, damit die Datei _makeJARRelease.bat auch die übrigen Dateien, die für Primerio notwendig sind dort hineinkopieren kann. Hier also das Verzeichnis ../Primerio-Licht12/binJar
- Weitere Einstellungen bitte unten stehender Abbildung entnehmen (Achtung! Die Verzeichnisstruktur ist die von Primerio, es geht um die anderen Einstellungen):



Dieses Bild ist aus dem neuen eclipse 3.0. (als Dateiname muss hier Primerolicht.jar stehen!)

- Dann auf **Next** klicken
- Der nächste Dialog benötigt i.d.R. keine weiteren Einstellungen, also **Next** klicken
- Im nächsten Dialog auf Browse klicken und den sich öffnenden Dialog mit OK bestätigen
- Die weiteren Einstellungen in diesem Dialog bitte unten stehender Abbildung entnehmen, bei main class muss jetzt Primerolicht stehen



- Dann auf **Finish** klicken. Das Jar wird erstellt und wenn kein weiterer Dialog erscheint ist alles OK.

Die Datei PrimeroLicht.jar befindet sich dann also im Verzeichnis ../Primero-Licht12/binJar.

Jetzt die Datei -makeJAR-Release.bat ausführen (ist im Verzeichnis Primero-Licht12), kopiert die fehlenden Dateien ins Verzeichnis /binJar.

Test z.B. mit einer der Beispieldateien (unter Primero-Licht12/Projekte). Ergebnis: es läuft! Mit der .jar -Datei kann dann (mit einem anderen Programm) eine komplette setup.exe erstellt werden.

Eigentlich kann der Start auch alternativ mit PrimeroLicht.exe (ebenfalls im Verzeichnis ../binJar) erfolgen.

Damit kann dann (mit einem anderen Programm) eine komplette setup.exe erstellt werden.

Der Ort der beiden nachträglich offen gelegten Dateien ist:
ALPrintDialog.java >> src/primero/packages/dialogs
Point3D.java >> src/plugin/lib/data

2 PRIMERO-Maxi

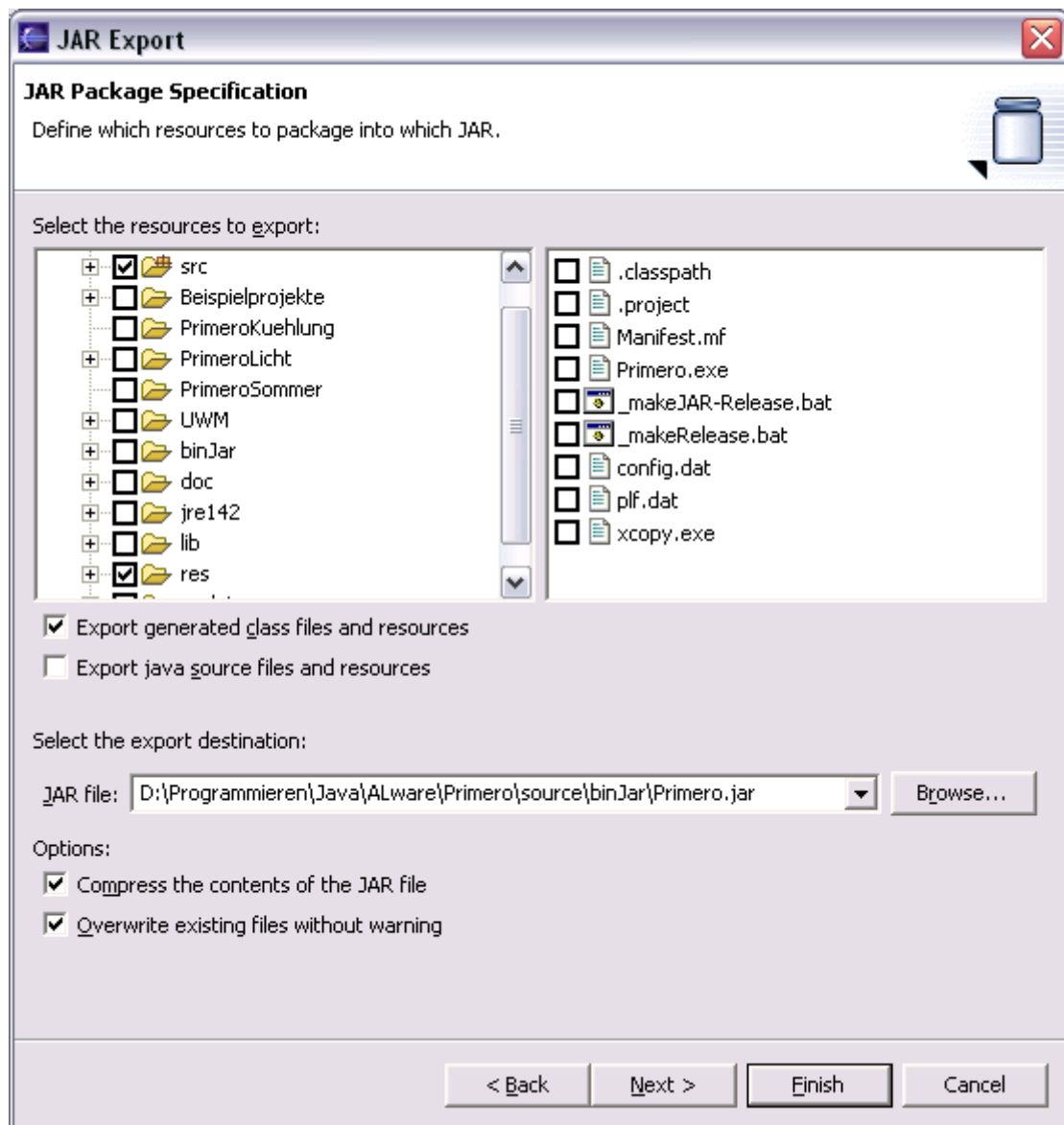
Voraussetzung: das Programm EnEV.exe ist bereits vorhanden! (Beschreibung zu diesem Programm weiter hinten!)

Falls notwendig, erst mal die richtige Ansicht in eclipse auswählen:
menü window >> open perspective >> java browsing

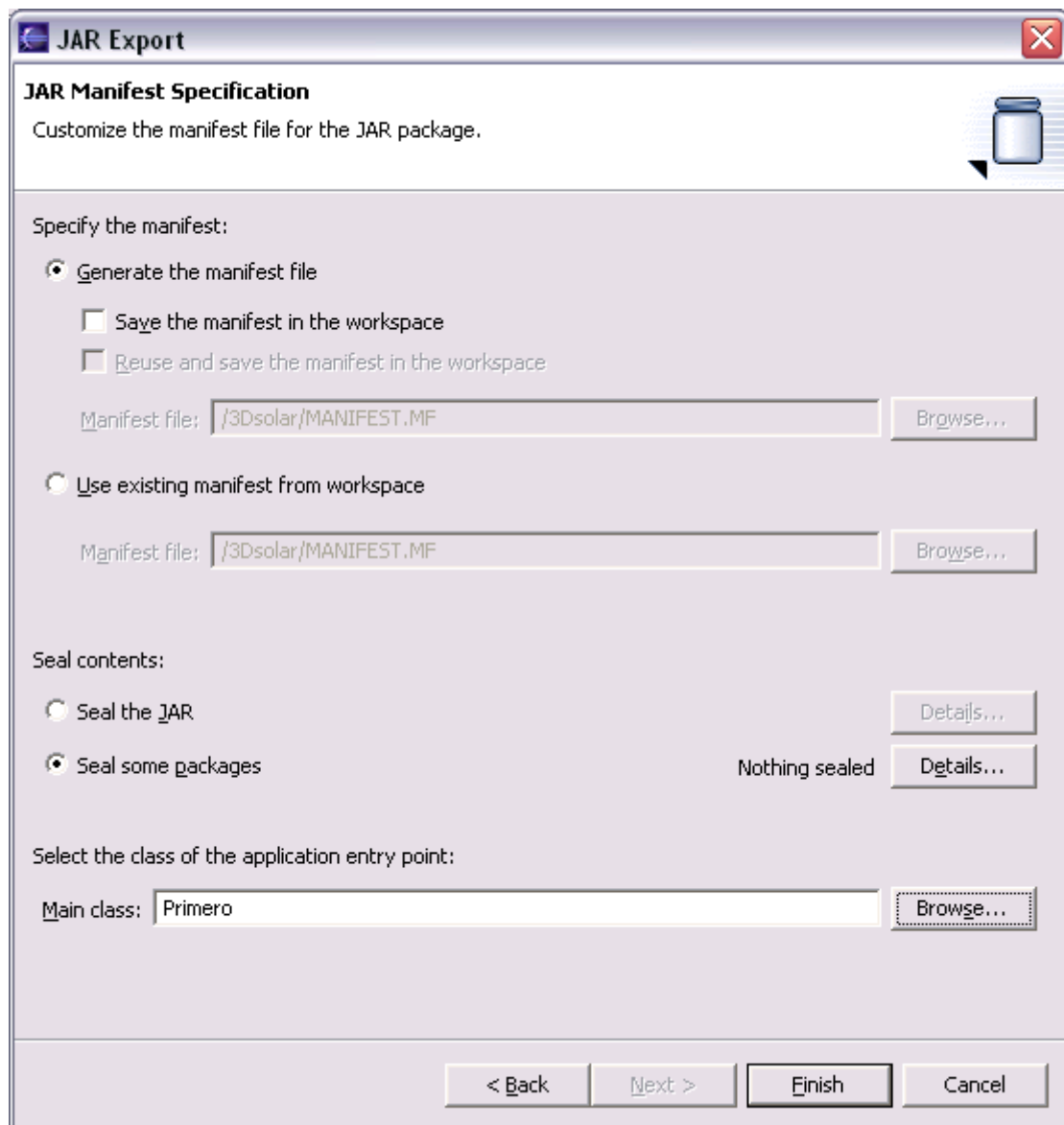
Das vorhandene Projekt in eclipse importieren
eclipse akzeptiert prinzipiell nur Projekte/Dateien/Quellcode, der über die Funktion importieren reingeholt wird.
dazu im Menü file >> import >> Existing Project into workspace next >> bei Project contents jetzt das Verzeichnis auswählen, in dem die Datei „project“ liegt (hier „Primero\Quellcode\source“) >> finish
Das Projekt wird dann geladen.

Erzeugen der Jar-Datei von Primero

- Im Fenster **Projects** mit der rechten Maustaste auf den Baumeintrag **Primero** klicken, dann im sich öffnenden Popup-Menü auf **Refresh** klicken
- Dasselbe danach mit dem Baumeintrag **src** vollziehen (befindet sich unterhalb des Eintrags **Primero**)
- Mit der rechten Maustaste auf den Eintrag **src** im Baum klicken und im Menü auf **Export...**
- Im aufgehenden Fenster Doppelklick auf **JAR file**
- Nun den Baum für **Select the resources for export...** aufklappen (Pluszeichen neben Eintrag **Primero**)
- Eintrag **src** ist schon markiert, so lassen, zusätzlich weiter unten noch ein Häkchen für Eintrag **res** setzen
- Mit dem Windows-explorer das Verzeichnis Primero/source/**binJar** anlegen (hier schon vorhanden)
- Nun noch den Pfad für die JAR-Datei angeben bei **JAR file:** **Wichtig** ist, dass dabei das Verzeichnis binJar ausgewählt wird, in dem sich das Projekt befindet, damit die Datei _makeJARRelease.bat auch die übrigen Dateien, die für Primero notwendig sind dort hineinkopieren kann. Hier das Verzeichnis ../source/binJar
- Weitere Einstellungen bitte unten stehender Abbildung entnehmen:



- Dann auf **Next** klicken
- Der nächste Dialog benötigt i.d.R. keine weiteren Einstellungen, also **Next** klicken
- Im nächsten Dialog auf Browse klicken und den sich öffnenden Dialog mit OK bestätigen
- Die weiteren Einstellungen in diesem Dialog bitte unten stehender Abbildung entnehmen



- Dann auf **Finish** klicken. Das Jar wird erstellt und wenn kein weiterer Dialog erscheint ist alles OK.

Die Datei Primero.jar befindet sich dann also im Verzeichnis ../source/binJar.

Jetzt die Datei -makeJAR-Release.bat ausführen (ist im Verzeichnis source), kopiert die fehlenden Dateien ins Verzeichnis /binJar.

Test z.B. mit den Projektbeispielen unter ../source/Projektbeispiele. Ergebnis: es läuft!

Der Start kann auch alternativ mit Primero.exe (ebenfalls im Verzeichnis ../binJar) erfolgen.

Damit kann dann (mit einem anderen Programm) eine komplette setup.exe erstellt werden.

3 Primero-Sommer DIN

Einzige Unterschiede hierbei sind:

- Die zu erzeugende JAR-Datei muss den Namen erhalten: PrimeroSommerDIN.jar
- Als Pfad der JAR-Datei muss natürlich das Verzeichnis binJar des Primero-Sommer DIN-Projektes angegeben werden
- Die ausgewählte main-class lautet: PrimeroSommerDIN

Falls notwendig, erst mal die richtige Ansicht in eclipse auswählen:
menü window >> open perspective >> java browsing

Das vorhandene Projekt in eclipse importieren

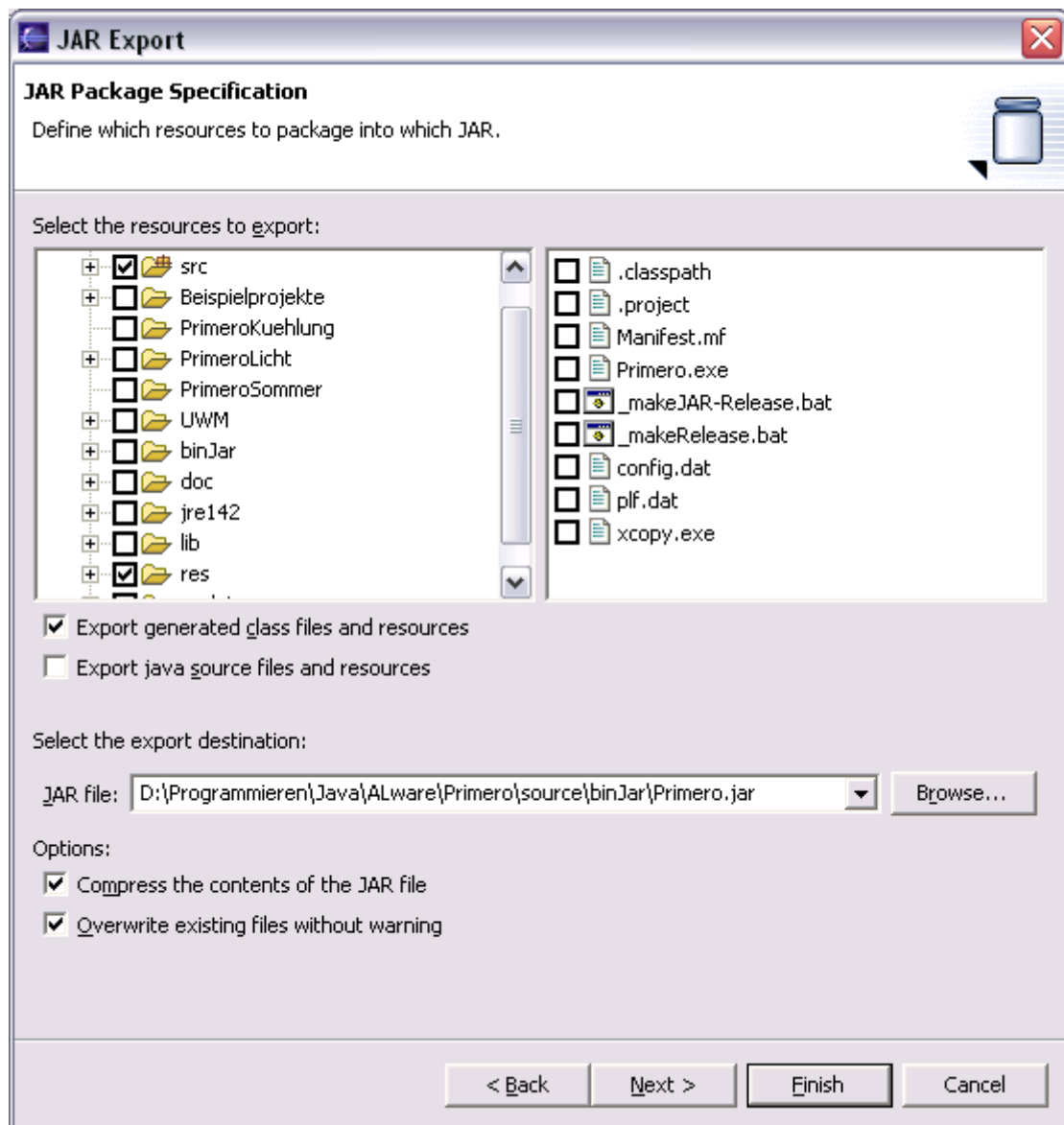
eclipse akzeptiert prinzipiell nur Projekte/Dateien/Quellcode, der über die Funktion importieren reingeholt wird.

dazu im Menü file >> import >> Existing Project into workspace next >> bei Project contents jetzt das Verzeichnis auswählen, in dem die Datei „project“ liegt (hier „Primero-Sommer-DIN\Quellcode\source“) >> finish

Das Projekt wird dann geladen.

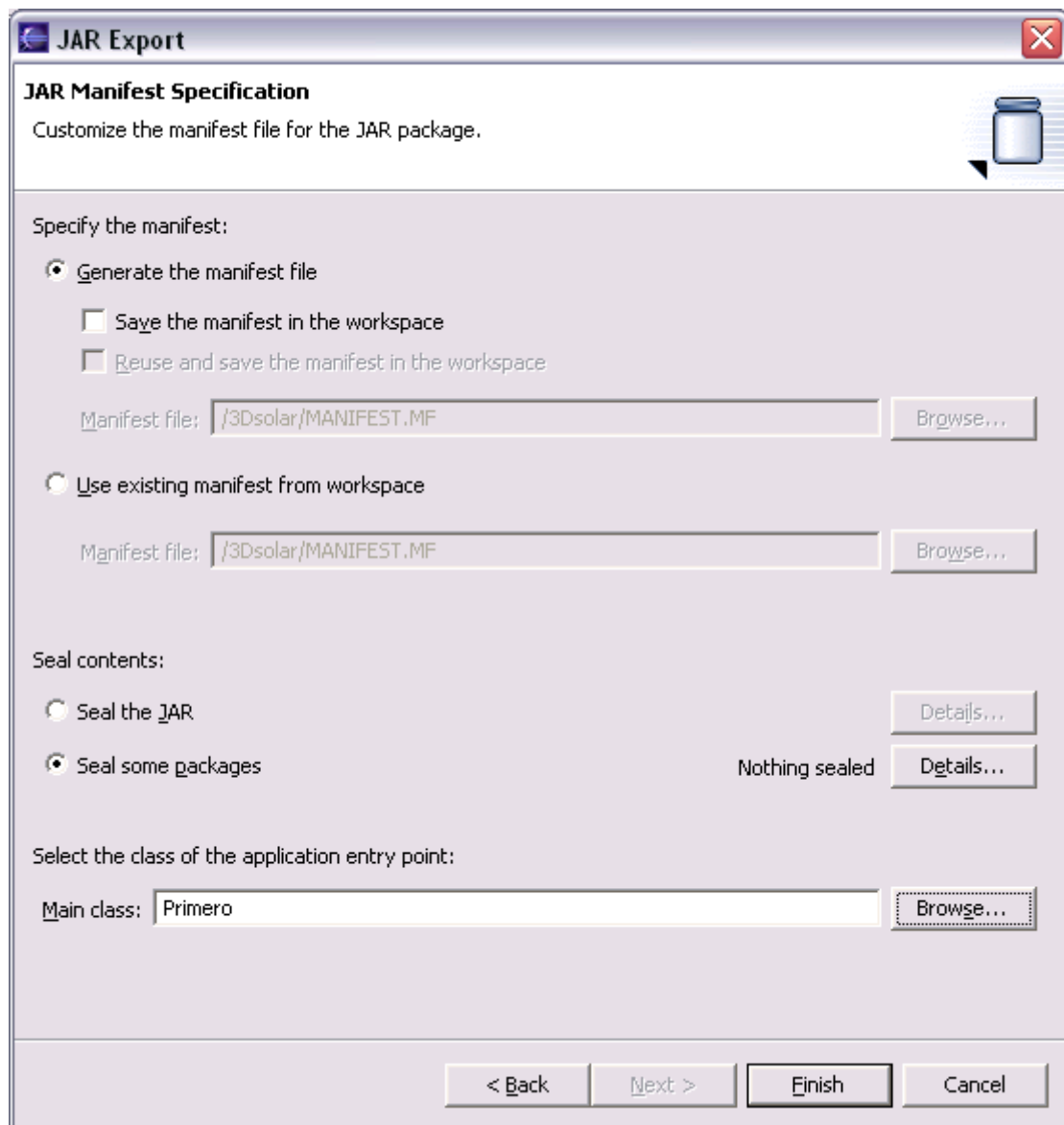
Erzeugen der Jar-Datei von Primero

- Im Fenster **Projects** mit der rechten Maustaste auf den Baumeintrag **Primero-Sommer-DIN** klicken, dann im sich öffnenden Popup-Menü auf **Refresh** klicken
- Dasselbe danach mit dem Baumeintrag **src** vollziehen (befindet sich unterhalb des Eintrags **Primero-Sommer-DIN**)
- Mit der rechten Maustaste auf den Eintrag **src** im Baum klicken und im Menü auf **Export...**
- Im aufgehenden Fenster Doppelklick auf **JAR file**
- Nun den Baum für **Select the resources for export...** aufklappen (Pluszeichen neben Eintrag **Primero-Sommer-DIN**)
- Eintrag **src** ist schon markiert, so lassen, zusätzlich weiter unten noch ein Häkchen für Eintrag **res** setzen
- Mit dem Windows-explorer das Verzeichnis Primero-Sommer-DIN/Quellcode/source/**binJar** anlegen (hier schon vorhanden)
- Nun noch den Pfad für die JAR-Datei angeben bei **JAR file:** **Wichtig** ist, dass dabei das Verzeichnis binJar ausgewählt wird, in dem sich das Projekt befindet, damit die Datei _makeJARRelease.bat auch die übrigen Dateien, die für Primero notwendig sind dort hineinkopieren kann. Hier das Verzeichnis ../source/binJar
- Weitere Einstellungen bitte unten stehender Abbildung entnehmen:



(Der screenshot ist von Primero-Maxi! Hier muss der Pfad jetzt auf Primero-Sommer-DIN.jar gestellt sein!)

- Dann auf **Next** klicken
- Der nächste Dialog benötigt i.d.R. keine weiteren Einstellungen, also **Next** klicken
- Im nächsten Dialog auf Browse klicken und den sich öffnenden Dialog mit OK bestätigen
- Die weiteren Einstellungen in diesem Dialog bitte unten stehender Abbildung entnehmen



(Der screenshot ist von Primero-Maxi! An Stelle von Primero muss hier Primero-Sommer-DIN stehen!)

- Dann auf **Finish** klicken. Das Jar wird erstellt und wenn kein weiterer Dialog erscheint ist alles OK.

Die Datei Primero-Sommer-DIN.jar befindet sich dann also im Verzeichnis ../source/binJar.

Jetzt die Datei -makeJAR-Release.bat ausführen (ist im Verzeichnis source), kopiert die fehlenden Dateien ins Verzeichnis /binJar.

Test z.B. mit den Projektbeispielen unter ../source/Projektbeispiele. Ergebnis: es läuft!

Der Start kann auch alternativ mit Primero-Sommer-DIN.exe (ebenfalls im Verzeichnis ../binJar) erfolgen.

Damit kann dann (mit einem anderen Programm) eine komplette setup.exe erstellt werden.

4 PLiG2

Das vorhandene Projekt in eclipse importieren

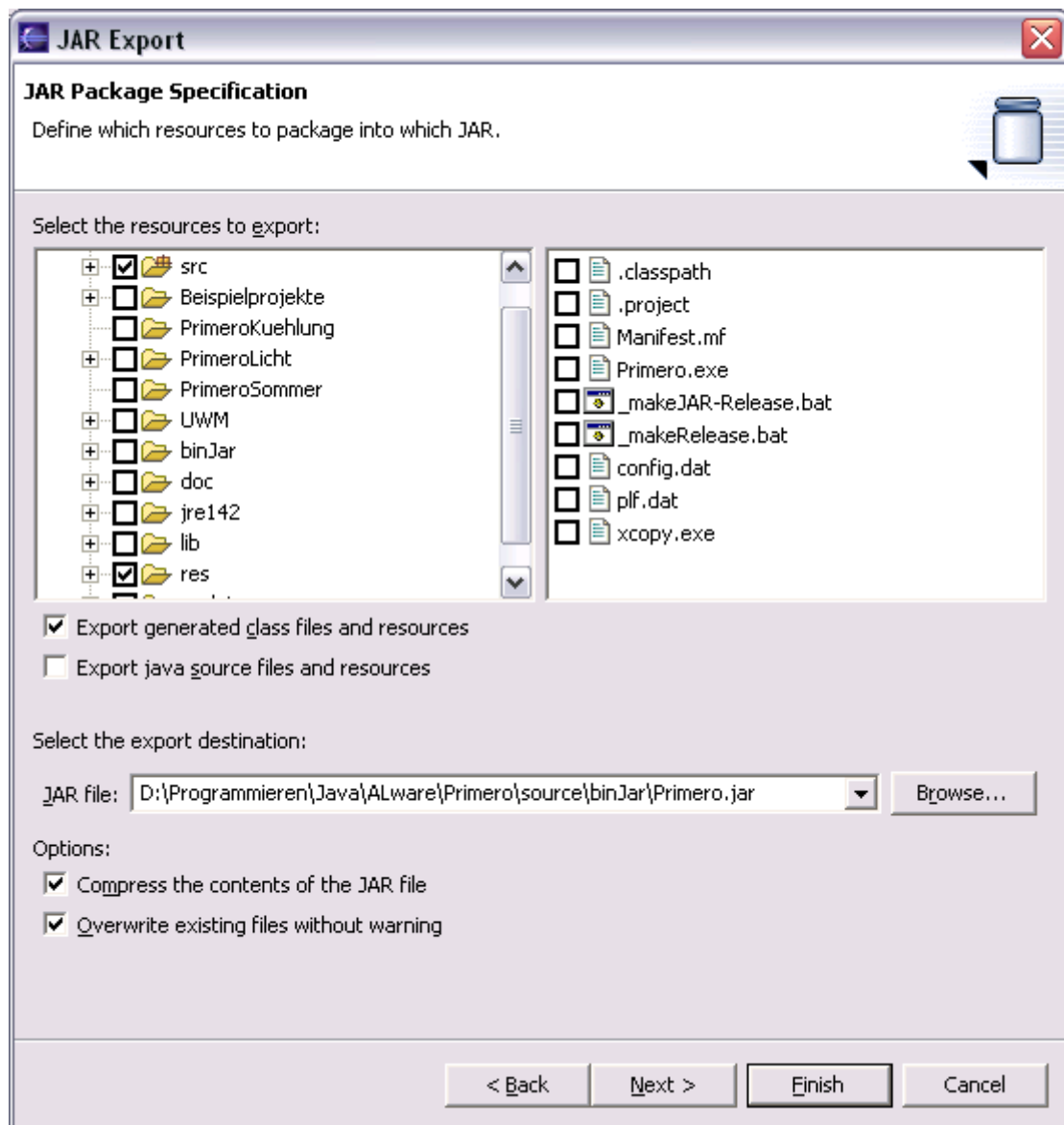
eclipse akzeptiert prinzipiell nur Projekte/Dateien/Quellcode, der über die Funktion importieren reingeholt wird.

dazu im Menü file >> import >> Existing Project into workspace next >> bei Project contents jetzt das Verzeichnis auswählen, in dem die Datei „project“ liegt (hier „PLiG2“) >> finish

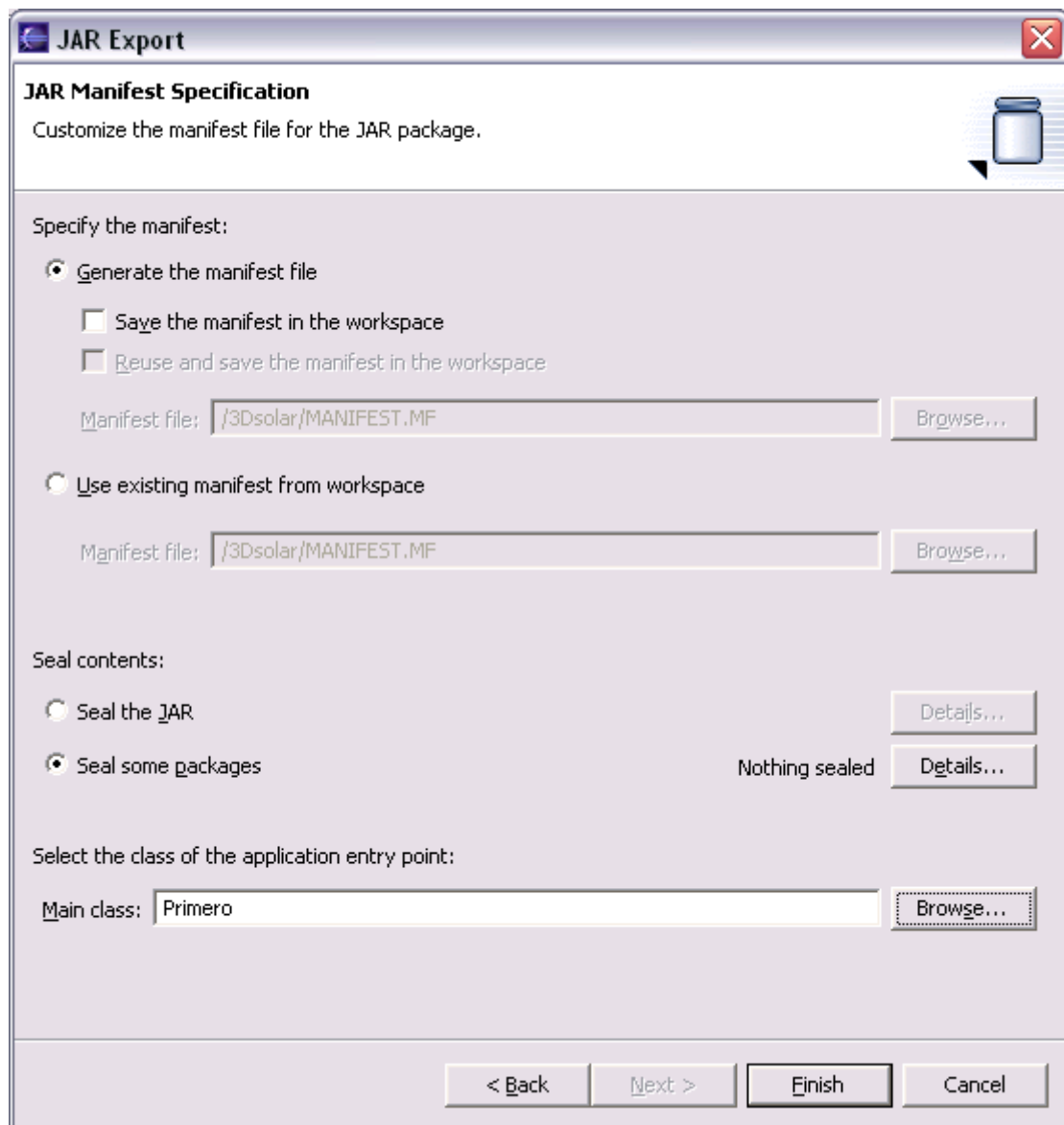
Das Projekt wird dann geladen.

Erzeugen der Jar-Datei von Primero-Licht

- Im Fenster **Projects** mit der rechten Maustaste auf den Baumeintrag **PLiG2** klicken, dann im sich öffnenden Popup-Menü auf **Refresh** klicken
- Dasselbe danach mit dem Baumeintrag **src** vollziehen (befindet sich unterhalb des Eintrags **PLiG2**)
- Mit der rechten Maustaste auf den Eintrag **src** im Baum klicken und im Menü auf **Export...**
- Im aufgehenden Fenster Doppelklick auf **JAR file**
- Nun den Baum für **Select the resources for export...** aufklappen (Pluszeichen neben Eintrag **PLiG2**)
- Eintrag **src** ist schon markiert, so lassen, zusätzlich weiter unten noch ein Häkchen für Eintrag **lib** und **res** setzen
- Mit dem Windows-explorer das Verzeichnis **PLiG2/binJar** anlegen (falls noch nicht vorhanden)
- Nun noch den Pfad für die JAR-Datei angeben bei **JAR file:** **Wichtig** ist, dass dabei das (soeben angelegte) Verzeichnis binJar ausgewählt wird, in dem sich das Projekt befindet, damit die Datei `_makeJARRelease.bat` auch die übrigen Dateien, die für PLiG2 notwendig sind dort hineinkopieren kann. Hier also das Verzeichnis `../binJar/PLiG2.jar`
- Weitere Einstellungen bitte unten stehender Abbildung entnehmen (Achtung! Die Verzeichnisstruktur ist die von Primero, es geht um die anderen Einstellungen):



- Dann auf **Next** klicken
- Der nächste Dialog benötigt i.d.R. keine weiteren Einstellungen, also **Next** klicken
- Im nächsten Dialog auf Browse klicken und den sich öffnenden Dialog mit OK bestätigen
- Die weiteren Einstellungen in diesem Dialog bitte unten stehender Abbildung entnehmen, bei main class muss jetzt PLiG2 stehen



- Dann auf **Finish** klicken. Das Jar wird erstellt und wenn kein weiterer Dialog erscheint ist alles OK.

Die Datei PLiG2.jar befindet sich dann also im Verzeichnis ../binJar.

Jetzt die Datei -makeJAR-Release.bat ausführen (ist im Verzeichnis PLiG2), kopiert die fehlenden Dateien ins Verzeichnis /binJar.

Test z.B. mit Aufruf der .jar Datei.

Eigentlich kann der Start auch alternativ mit PLiG2.exe (ebenfalls im Verzeichnis ../binJar) erfolgen.

Compilieren der Borland Delphi – Quellcodes

5 DL_frac

Alle von Detlef Hennings gelieferten Dateien (CD) sind im Verzeichnis DL_frac2.
Am besten verwendet man für den Quellcode die Dateien von der Diskette und nicht die von Detlefs CD. Mit den Daten von der CD wird einmal das Programm DL_fra2.exe erzeugt an Stelle DL_frac.exe und zweitens ist noch ein Testmodus aktiviert, der jedes mal den Compiler startet. Mit den Daten von der Diskette geht es ganz direkt...

Die Projektarbeitsmappe ist DL_frac.dpr – einfach doppelklicken oder vom Compiler aus öffnen.

Starten des Compilers mit dem Menü „Start“ und noch mal „Start“. Erzeugt sofort DL_frac.exe im selben Verzeichnis. Zum Test sind noch die Dateien .dif und DL_frac.in drin – es läuft!

NOCH NICHT GETESTET IST, OB DIESE DL_FRAC AUCH IN P_LICHT LÄUFT!!!
Es tut, aber mit einer Fehlermeldung „Datei DL_frac.io nicht gefunden“ wo kommt das her?

Compilieren der C++ – Quellcodes

6 BAX

Der Quellcode ist auf der Beta CD bisher nicht drauf, aber liegt an der HAW vor (Version 1.2).

Einfach auf der Datei BAX.sln doppelklicken, öffnet den Compiler.

Oben in der Abrollbox „release“ einstellen.

Rechts im Projektmappenexplorer „BAX“ markieren.

Compilieren mit „Erstellen >> BAX erstellen“. Das Programm BAX.exe befindet sich dann im Unterverzeichnis \release.

Test: Einfügen (Ersetzen) des Programms in das Verzeichnis

c:\PRIMERO_ENTWICKLUNG\Primero-Licht\lib\bax, Ergebnis: es läuft!

BAX dient ja als Windows-Startprogramm für die diversen jar-Dateien. BAX erhält dann jeweils den Namen des Programms, also z.B. Primero.exe, Primero-Licht.exe usw. und mit dem ResourceHacker kann man auch das Icon ändern.

Diesen einfach öffnen, die zu hackende exe-Datei laden (File->Open) und dann im Baum den Eintrag "Icon" aufklappen und dort den ersten Untereintrag aufklappen.

Mit linker Maustaste das Icon markieren, danach mit rechter Maustaste anklicken und im Popup-Menü den Eintrag "Replace Resource" auswählen.

Dann in den sich öffnenden Dialog den Knopf "Open file with new icon" anklicken. Ein Icon auswählen und OK drücken.

Dann im Dialog auf Knopf "Replace" klicken.

Danach im Hauptfenster im Menu File->Save die neue Exe-Datei speichern.

Das neue Icon wird erst aktualisiert wenn der Explorer neu aufgemacht wird, also nicht wundern wenn das Icon nicht sofort sichtbar ist. Im Zweifelsfall Rechner neu booten, dann sollte es auf jeden Fall angezeigt werden.

7 UWM

Einfach auf der Datei UWertManager.sln (in UWM\Version3.3\Quellcode) doppelklicken, öffnet den Compiler.

Compilieren mit „Erstellen >> UWertManager erstellen“.

Hinweis: beim Anlegen des Projektes im Compiler diese Einstellungen wählen:

- VISUAL C++ Projekte
- MFC Anwendung
Bei Anwendungstyp dann
- MFC in einer statischen Bücherei verwenden
- Einfaches Dokument
- MFC Standard

Das Programm UWertManager.exe befindet sich dann im Unterverzeichnis \release.

Test: Kopieren UWertManager.exe z.B. in ein Verzeichnis /Test. Dort drin müssen dann auch die Datenbanken Material_DIN4108.dat und Material_User.dat sein und als Unterverzeichnisse /res und /hilfe - Ergebnis: es läuft!

Compilieren der FORTRAN – Quellcodes

8 EnEV

Zuerst den Compiler zu Fuß starten, z.B. über den Startknopf von Windows (Compaq Visual Fortran 6 >> Developer Studio).

File >> open workspace >> EnEV.dsw auswählen (im Verzeichnis EnEV)

Einstellen *build >> set active configuration >> release*

Compilieren und linken mit *build >> build EnEV.exe*

Die Datei EnEV.exe ist dann im Verzeichnis ../release.

Test: einfach starten, im Verzeichnis befindet sich ein Satz Inputdateien. Die Ausgabedateien werden erzeugt.

Oder: ersetzen in Primero/source/binJar/lib/modules - Ergebnis: es läuft!

9 Lichtdll

Ziel ist das Erstellen der Datei LichtDLLJava.dll mit dem C++ Compiler. Nur so eine dll kann von Java verstanden werden.

Schritt 1

Dazu wird zunächst der für eine dll variierte Fortran Quellcode Lichtdll.for compiliert. Den Compiler zu Fuß starten, z.B. über den Startknopf von Windows (Compaq Visual Fortran 6 >> Developer Studio).

File >> open workspace >> LichtDLLFortran.dsw auswählen (im Verzeichnis Lichtdll/LichtDLLFortran)

Einstellen *build >> set active configuration >> release*

Compilieren und linken mit *build >> build LichtDLLFortran.dll*

Den Fortran-Compiler wieder schließen!

Schritt 2

Das C++ Projektverzeichnis ist LichtDLLneu. Dort hineinkopieren: alle 4 Dateien, die in LichtDLLFortran\release stehen

Schritt 3

Öffnen von LichtDLLneu.sln mit dem C++ Compiler (einfach doppelklicken).

Auswählen der Ansicht am besten Ansicht >> Projektmappenexplorer.

Die mit dem FORTRAN-Compiler erzeugten Dateien „LichtDLLFortran.lib“ und „lichtdll.obj“ rechts im Projektmappenexplorer hinzufügen mit:
Markieren des Verzeichnisses „lichtdllneu“, rechte Maustaste, Auswahl hinzufügen
>> vorhandenes Element hinzufügen, Auswahl der Datei.

Erstellen der dll unter C++ mit „Erstellen“ >> „Lichtdllneu erstellen“. Die fertige dll liegt jetzt im Verzeichnis „release“ und hat den Namen „LichtDLLJava.dll“.

Test: Ersetzen der dll im Verzeichnis

Primero_Entwicklung\PrimeroLicht\source\binJar\lib\licht (wenn das Lichtprogramm solo laufen soll) UND im Verzeichnis

Primero_Entwicklung\Primero\source\binJar\PrimeroLicht\lib\licht (wenn das Lichtprogramm unter Primero-Maxi laufen soll).

Ergebnis: es läuft!