

Allgemeine Schwerpunkte der Arbeit

- Recherche zu Automatisierungsmöglichkeiten (APIs und Tools) von PTC Creo Elements/Pro 5.0
- Vergleich der APIs von PTC (Pro/TOOLKIT, Object Toolkit, J-Link, Pro/Web.Link, Visual Basic API)
- Programmieren einer Importanwendung inkl. Dokumentation

Praxis und Ziel im Klartext

Erzeugen einer Anwendung, die automatisiert die von MOCAD generierten Kurvenkonturen im .ibl-Dateiformat inklusive Getriebeparametern nach Creo Elements/Pro 5.0 importiert und diese zu ebenen Kurvenkörpermodellen extrudiert

Umsetzung und Ergebnis

Verwendete Software

- Microsoft Windows 7 Professional
- Netbeans IDE 7.2 Beta
- Java JDK 1.7.0_02-b13
- PTC Creo Elements/Pro 5.0 M130



Endergebnis

- Entwicklung eines Plug-Ins mit Schnittstelle J-Link
- Namenstaupe des Plug-Ins → „MoTract“

Screenshot – Konsole (Commandline Input)

```
Creo-Startbefehl: proe1
Einklinken in aktive Sitzung: false
Konfigurationsdatei: C:\Program Files\PTC\Creo Elements\Pro5.0\Creo\URZ_Konfi
ation\Einstellungen\config.pro
IBL-Datei: Example\test1.ibl
Modelldatei: Example\moc_kurvenscheibe.prt
Console-Output in LogFile umleiten? false
Creo nach Import/Update beenden? false
Stiller Modus? false
TimeOut-Maximum = 10 Sekunden
```

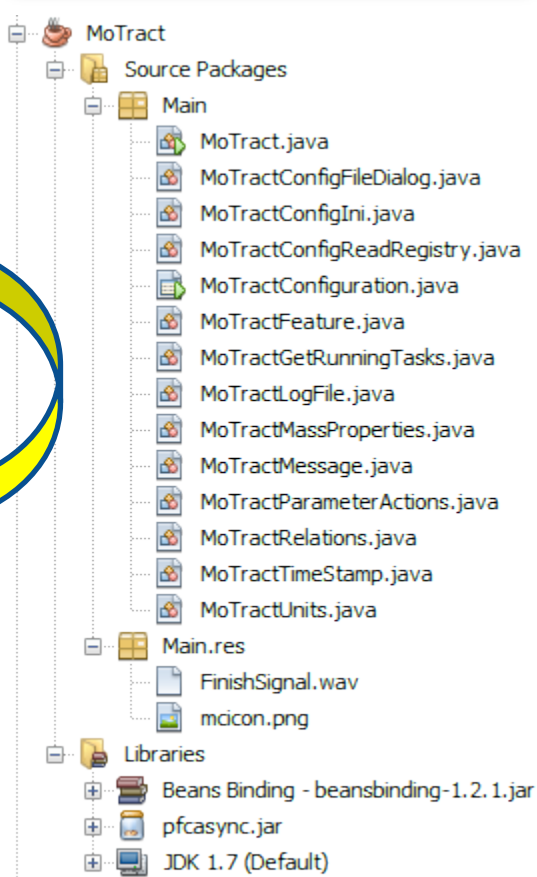
```
Thread RuntimeThread = new Thread(new Runnable()
{
@Override
public void run()
{
try
{
if (MoTractSilentModeParsed == false)
{
MoTractRuntime = pfcAsyncConnection.AsyncConnection_Start (MoTractSta
}
else
{
MoTractRuntime = pfcAsyncConnection.AsyncConnection_Start (MoTractSta
}

if ("updatecamdisk".equals(MoTractJobToDo.toLowerCase()))
{
MoTractJobCase = "Update Modelldatei";
MoTractUpdateCamDisk();
}
else
```

Java und J-Link
→ Parametric Foundation Classes



Projektverzeichnis – Netbeans IDE



Screenshot – Benutzeroberfläche

