

Siemens  
PLM Connection



Deutschland 2008

## Vortragstitel

Nx Massendatenverarbeitung im Teamcenter Umfeld (JobManagement)

Summery:

Massendaten Verarbeitung im NX/TcEng Umfeld.

Am Beispiel „JT File Erzeugung“ und automatisiertem Datenexport.

Refferent:

Josef Feuerstein

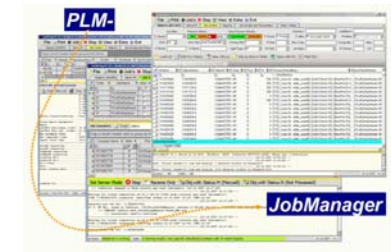
Dipl.Ing.J.Feuerstein, Hauptstr. 6, 36142 Tann

[Josef.Feuerstein@PLMJobManager.COM](mailto:Josef.Feuerstein@PLMJobManager.COM)

Tel.: 06682-97060



Benutzergruppe e.V.





# Inhaltsverzeichnis

<a href="#"><u>Hinweise zur der Präsentation</u></a>	Seite: 3
<a href="#"><u>Was ist ein JT File</u></a>	Seite: 4
<a href="#"><u>Wie werden JT Files verwendet</u></a>	Seite: 5
<a href="#"><u>Welche Methoden gibt es JT Files zu erzeugen</u></a>	Seite: 6
<a href="#"><u>Beispiel JT File erzeugung aus NX heraus</u></a>	Seite: 7
<a href="#"><u>Die Einstellungen um JT Files beim Arbeiten mit NX zu speichern</u></a>	Seite: 8
<a href="#"><u>Welche Einstellungen wirken zusammen</u></a>	Seite: 9
<a href="#"><u>Arbeiten mit ugotpv und Scripten</u></a>	Seite: 10
<a href="#"><u>Beispiel 1 Export Einzelteil in ein Verzeichniss</u></a>	Seite: 11
<a href="#"><u>Beispiel 2 Export Einzelteile in ein Verzeichniss als Input wird eine Datei angegeben</u></a>	Seite: 12
<a href="#"><u>Beispiel 3 Einzelteile im Teamcenter aktualiesieren</u></a>	Seite: 13
<a href="#"><u>Beispiel 4 Export Baugruppe Monolitish in ein Verzeichniss</u></a>	Seite: 14
<a href="#"><u>Beispiel 5 Export Baugruppe mit allen Teilen in ein Unterverzeichniss</u></a>	Seite: 15
<a href="#"><u>Beispiel 6 Baugruppe Monolitish für TCE erzeugen</u></a>	Seite: 16
<a href="#"><u>Get Help</u></a>	Seite: 17
<a href="#"><u>Wie können solche Vorgänge insgesamt automatisiert werden</u></a>	Seite: 18
<a href="#"><u>Die Oberfläche des PLMJobManager</u></a>	Seite: 19
<a href="#"><u>PLMJobManager Site Setup und Vorbereitung der Datenbanken</u></a>	Seite: 20
<a href="#"><u>PLMJobManager Steuerung der NX-Refile Optionen und Scripte</u></a>	Seite: 21
<a href="#"><u>Arbeiten mit dem JobServer Job Control</u></a>	Seiten: 22-23

## Hinweise zur der Präsentation

**Alle hier vorgestellten Folien beziehen sich auf den Einsatz von:**

**„NX4 in Verbindung mit TCE9.1.3.0“**

**Hinweis:**

**Dieser Vortrag ist kein spezial Vortrag zum Thema „JT File Erstellung“. Anspruch dieses Vortrages ist es die verschieden Methoden für Massendatenverarbeitung am Beispiel JT's Exemplarisch zu zeigen.**



# Was ist ein JT File

## *Auszug aus dem CAD-Lexikon*

02.09.03

### **Begriff Jupiter Tessellation**

**Erläuterung** JT (= Jupiter Tessellation) ist eine Technologie zur Erzeugung von Visualisierungsmodellen in einem kompakten, aber inhaltsreichem Datenformat. Bei dem Format DirectModel- bekannt unter dem Namen JT- handelt es sich um einen Quasistandard in der Automobil- und Luftfahrtindustrie für die Visualisierung von 3D-Geometrien, sowie deren Analyse und Optimierung. JT kann über einen Translationsprozess aus vielen CAD-Systemen generiert werden. Das Datenmodell unterstützt unterschiedliche Repräsentationen der CAD-Geometrie: von einfachen Dreiecksflächen bis hin zur filigranen Brep-Geometrie. Auch Struktur- und Metadaten werden von JT unterstützt.

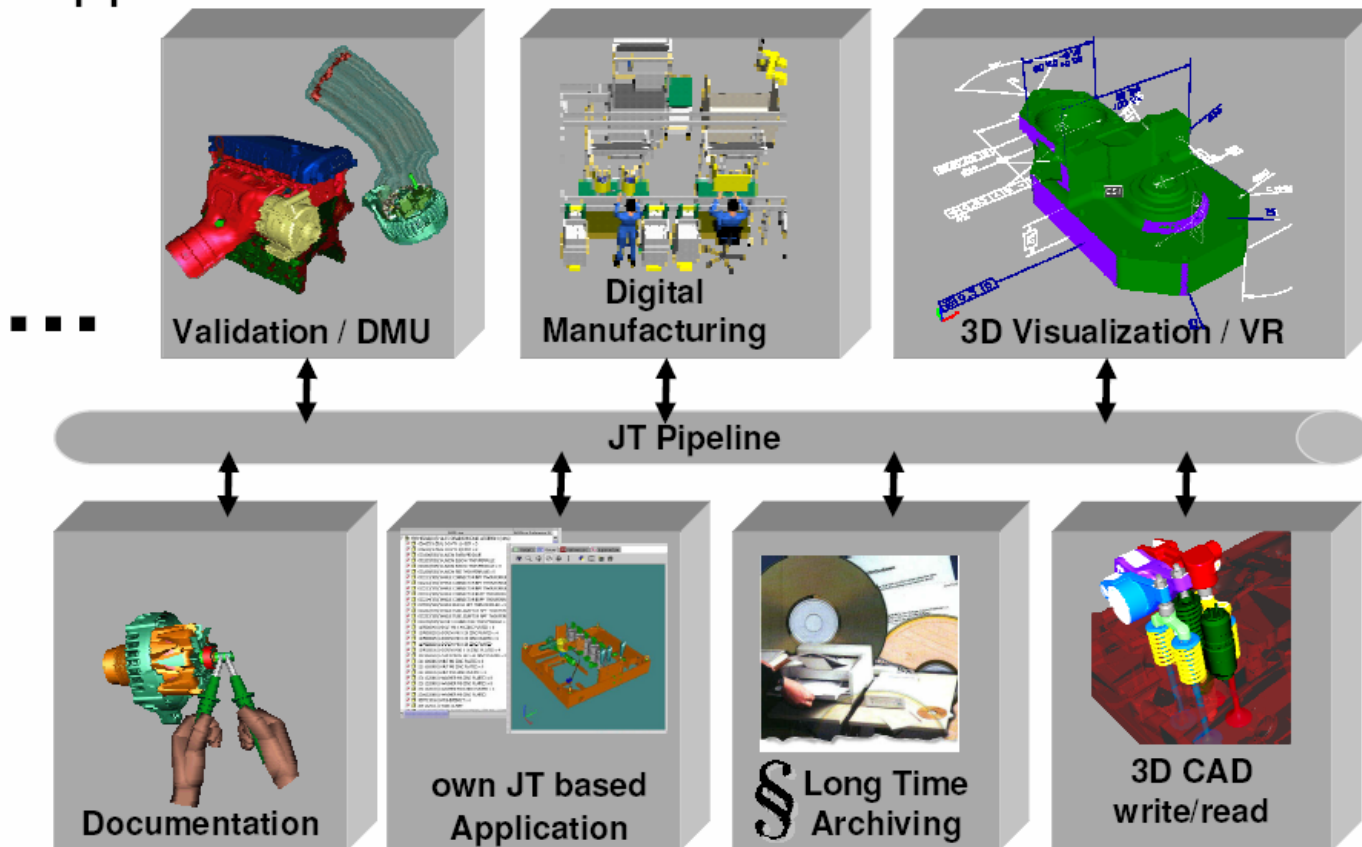
IT-Open ([www.jtopen.com](http://www.jtopen.com)) ist eine Interessensgemeinschaft von Endanwendern, Software-Herstellern, akademischen Partnern und unterstützenden Organisationen zur Förderung der JT-Technologie.



# Wie werden JT Files verwendet

## Current Status of Digital Engineering Collaboration

### Application Fields for JT



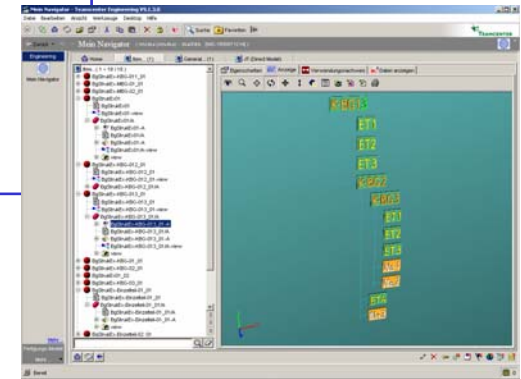
# Welche Methoden gibt es JT Files zu erzeugen

Werkzeuge:

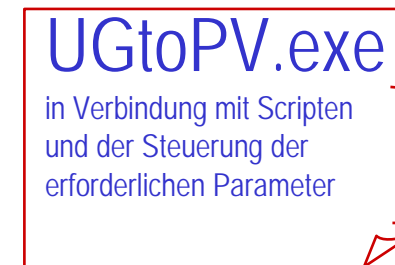
Ziel:



Können beim Speichern der NX Parts erzeugt werden



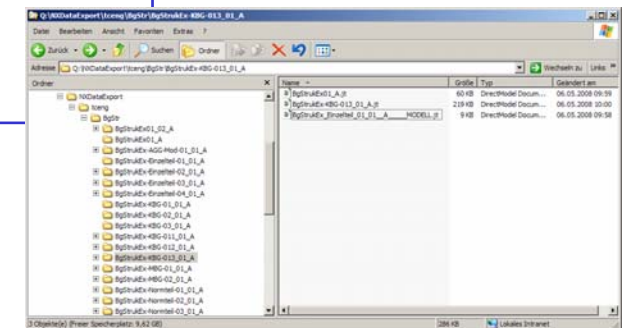
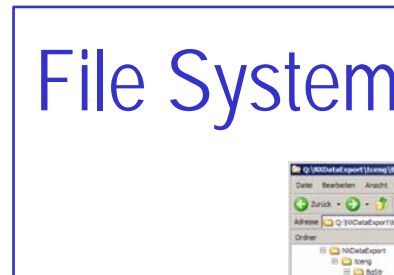
Erstellung über:  
Datei → Exportieren → JT..



in Verbindung mit Scripten und der Steuerung der erforderlichen Parameter

Erstellung durch nicht-verwenden des Parameters:  
-save\_outside\_iman

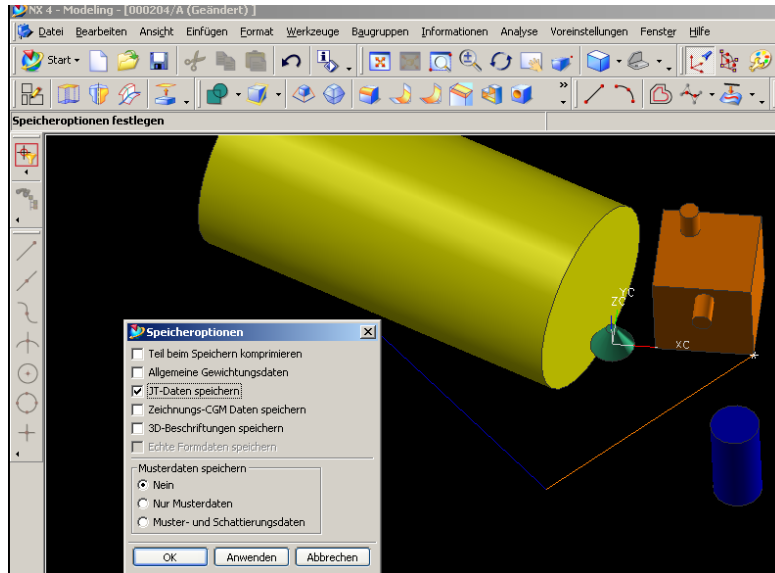
Erstellung mit Hilfe des Parameters:  
-save\_outside\_iman





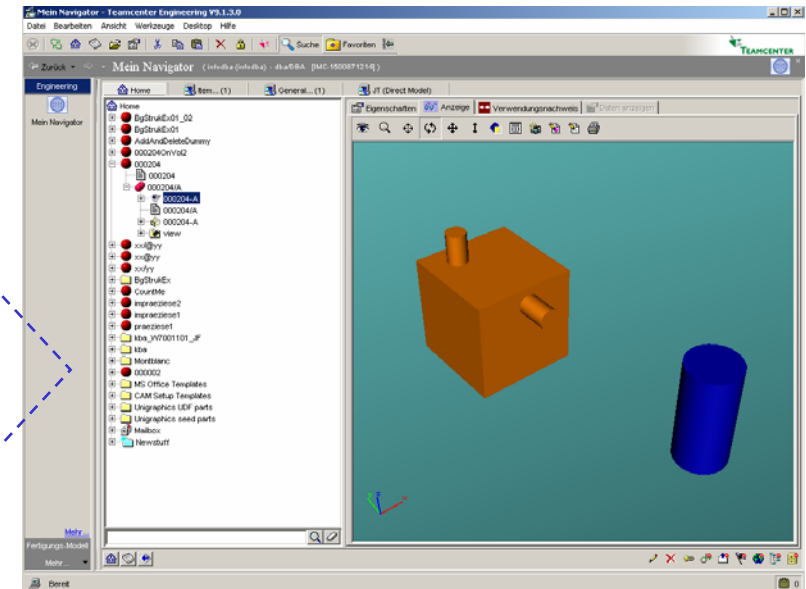
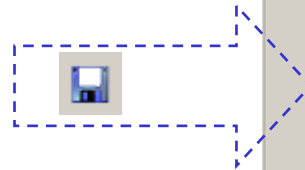
# Beispiel JT File erzeugung aus NX heraus

**JT in TCE erzeugen lassen:  
Datei → Optionen → Speicheroptionen**



Ergebniss:

JT File in TCE

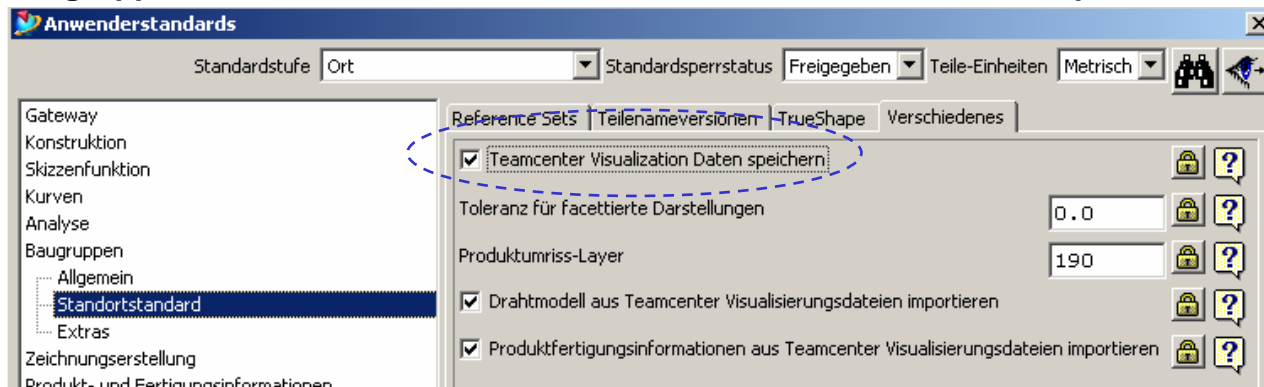


Hinweis:

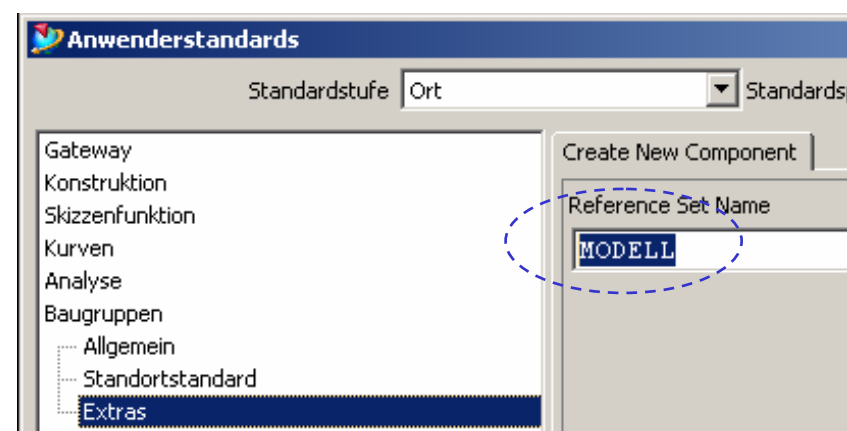
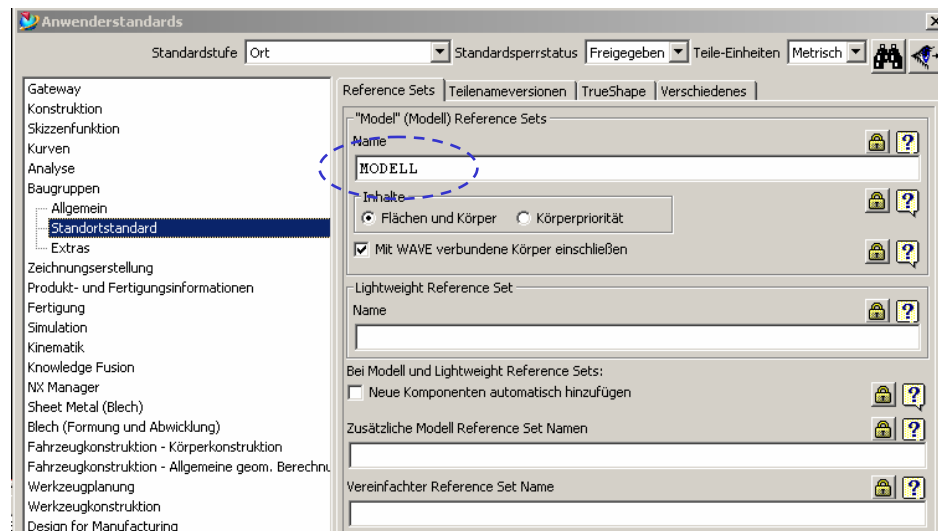
Welche Objekte in den JT File übertragen werden ist abhängig von den Voreinstellungen für Refsets und Layer. Und der Objekt Eigenschaften für Layer und Refsets

# Die Einstellungen um JT Files beim Arbeiten mit NX zu speichern

Sollen im Unternehmen beim speichern der NX Parst auch die JT Files gespeichert werden  
 So können diese über die Anwenderstandards voreingestellt werden  
 Baugruppen → Standortstandard → Teamcenter Visualization Daten speichern



Die hier folgenden Einstellungen müssen ebenfalls für das Unternehmen eingestellt werden



Weiter konfigurationen sind über die tessUG.config einstellbar.





## Welche Einstellungen wirken zusammen

set UGII\_LOCAL\_USER\_DEFAULTS=%UGII\_SITE\_DIR%\startup\Nx4\_site.dpv

Anwenderstandards

set UGII\_LOAD\_OPTIONS=%UGII\_SITE\_DIR%\load\_options.def

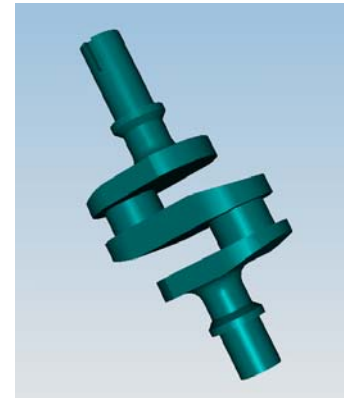
load\_options.def

set UGII\_PV\_TESS\_CONFIG\_FILE=%UGII\_SITE\_DIR%\tessUG.config

tessUG.config

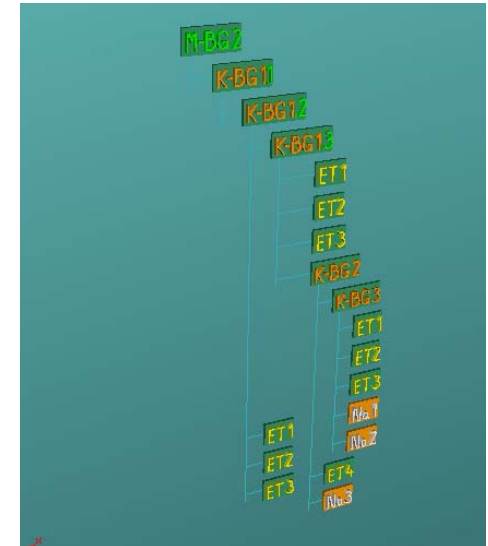
"%UGII\_BASE\_DIR%\PVTRANS\ugtopv.exe"

Ugtopv.exe  
Optionsparameter



Wird ein Einzelteil  
ausgegeben?

Wird eine Baugruppe ausgegeben?





## Arbeiten mit ugotpv und Scripten

- Script Shema:
- TcePrompt durchlaufen
- Setzen der Umgebung
- Setzen der Parameter
- Start des Programmes

Hinweis:

Es folgen Beispiele wie JT Daten mit Hilfe von Scripten erzeugt werden können



## Beispiel 1 Export Einzelteil in ein Verzeichniss

```
rem -----
rem ***** Beispiel 1 *****
rem -----
rem Beispiel: Export Singlepart in ein Verzeichniss
rem Part:=@DB/BgStrukEx-Einzelteil-01_01/A
rem Ergbniss: obriges Einzelteil wird exportiert

set tessConfig_DPNE=%CurCmdRoot_DP%\tessUG.Config

set toRun="%UGII_BASE_DIR%\PVTRANS\ugtopv.exe"
set toRun=%toRun% @DB/BgStrukEx-Einzelteil-01_01/A
set toRun=%toRun% -pim=yes -u=%TceUser% -p=%TceUserPassword%

set JobParameter=-nouupdate
set JobParameter=%JobParameter% -config="%tessConfig_DPNE%"
set JobParameter=%JobParameter% -save_outside_iman -force_output_dir="%JTFile_LP%"
set JobParameter=%JobParameter% -single_part

rem command vollständig zusammensetzen
set toRun=%toRun% %JobParameter%
rem goto :startJTGen
```



## Beispiel 2 Export Einzelteile in ein Verzeichniss als Input wird eine Datei angegeben

```
rem -----
rem ***** Beispiel 2 *****
rem -----
rem Beispiel: Export Einzelteile in ein Verzeichniss
rem Part[s] aus ListenDatei lesen:=%CurCmdRoot_DP%\InputfileTest2.txt
rem Achtung: !! aufbau der Inhalte der Datei: @DB/BgStrukEx-Einzelteil-01_01/A
rem Ergbniss: alle Teile der Liste werden exportiert

set tessConfig_DPNE=%CurCmdRoot_DP%\tessUG.Config

set toRun="%UGII_BASE_DIR%\PVTRANS\ugtopv.exe"
set toRun=%toRun% "%CurCmdRoot_DP%\InputfileTest2.txt"
set toRun=%toRun% -pim=yes -u=%TceUser% -p=%TceUserPassword%

set JobParameter=-noupdate
set JobParameter=%JobParameter% -config="%tessConfig_DPNE%"
set JobParameter=%JobParameter% -save_outside_iman -force_output_dir="%JTFile_LP%"
set JobParameter=%JobParameter% -single_part

rem command vollständig zusammensetzen
set toRun=%toRun% %JobParameter%

Echo -----
Echo Liste der zu Exportierenden Parts:
type "%CurCmdRoot_DP%\InputfileTest2.txt"

goto :startJTGen
```



## Beispiel 3 Einzelteile im Teamcenter aktualisieren

```
rem -----
rem ***** Beispiel 3 *****
rem -----
rem Beispiel: Einzelteile im Teamcenter aktualisieren
rem Part[s] aus ListenDatei lesen:=%CurCmdRoot_DP%\InputfileTest2.txt
rem Achtung: !! aufbau der Inhalte der Datei: @DB/BgStrukEx-Einzelteil-01_01/A
rem Ergbniss: alle Teile der Liste werden exportiert

set tessConfig_DPNE=%CurCmdRoot_DP%\tessUG.Config

set toRun="%UGII_BASE_DIR%\PVTRANS\ugtopv.exe"
set toRun=%toRun% "%CurCmdRoot_DP%\InputfileTest2.txt"
set toRun=%toRun% -pim=yes -u=%TceUser% -p=%TceUserPassword%

set JobParameter=-nouupdate
set JobParameter=%JobParameter% -config="%tessConfig_DPNE%"
set JobParameter=%JobParameter% -single_part

rem command vollständig zusammensetzen
set toRun=%toRun% %JobParameter%

Echo -----
Echo Liste der zu Parts:
type "%CurCmdRoot_DP%\InputfileTest2.txt"

goto :startJTGen
```



## Beispiel 4 Export Baugruppe Monolitisch in ein Verzeichniss

```

rem -----
rem ***** Beispiel 4 *****
rem -----
rem Beispiel: Export Baugruppe Monolitisch in ein Verzeichniss
rem Part:@DB/BgStrukEx01/A
rem Hinweis: hierfuer muss der erforderliche Parameter -honour_structure engesetzt werden
rem          dieser schliesst aus das ein inputfile eingesetzt wird!
rem Ergebniss: obrisges Einzelteil wird exportiert

rem Hinweis in dieser Config ist angepasst: structureOption = "MONOLITHIC"
set tessConfig_DPNE=%CurCmdRoot_DP%\tessUG.Config

set toRun="%UGII_BASE_DIR%\PVTRANS\ugtopv.exe"
set toRun=%toRun% @DB/BgStrukEx01/A
set toRun=%toRun% -pim=yes -u=%TceUser% -p=%TceUserPassword%

set JobParameter=-nouupdate
set JobParameter=%JobParameter% -config="%tessConfig_DPNE%"
set JobParameter=%JobParameter% -save_outside_iman -force_output_dir="%JTFile_LP%"
set JobParameter=%JobParameter% -honour_structure

rem command vollständig zusammensetzen
set toRun=%toRun% %JobParameter%
goto :startJTGen |

```



## Beispiel 5 Export Baugruppe mit allen Teilen in ein Unterverzeichniss

```
rem -----
rem ***** Beispiel 5 *****
rem -----
rem Beispiel: Export Baugruppe mit allen Teilen in ein Unterverzeichniss
rem Part:=@DB/BgStrukEx01/A
rem Hinweis: hierfuer muss der erforderliche Parameter -honour_structure engesetzt werden
rem          dieser schliesst aus das ein inputfile eingesetzt wird!
rem Ergebniss: obriges Einzelteil wird exportiert

rem Hinweis in dieser Config ist angepasst: structureOption = "PER_PART"
set tessConfig_DPNE=%CurCmdRoot_DP%\tessUGExportBgMitAllenTeilen.Config

set toRun="%UGII_BASE_DIR%\PVTRANS\ugtopv.exe"
set toRun=%toRun% @DB/BgStrukEx01/A
set toRun=%toRun% -pim=yes -u=%TceUser% -p=%TceUserPassword%

set JobParameter=-nouupdate
set JobParameter=%JobParameter% -config="%tessConfig_DPNE%"
set JobParameter=%JobParameter% -save_outside_iman -force_output_dir="%JTFile_LP%"
set JobParameter=%JobParameter% -honour_structure

rem command vollständig zusammensetzen
set toRun=%toRun% %JobParameter%
goto :startJTGen
```



## Beispiel 6 Baugruppe Monolitisch für TCE erzeugen

```
rem -----
rem ***** Beispiel 6 *****
rem -----
rem Beispiel: Baugruppe Monolitisch für TCE erzeugen
rem Part:@DB/BgStrukEx01/A
rem Hinweis: hierfuer muss der erforderliche Parameter -honour_structure engesetzt werden
rem          dieser schliesst aus das ein inputfile eingesetzt wird!
rem Ergebniss: obriges Einzelteil wird exportiert

rem Hinweis in dieser Config ist angepasst: structureOption = "MONOLITHIC"
set tessConfig_DPNE=%CurCmdRoot_DP%\tessUG.Config

set toRun="%UGII_BASE_DIR%\PVTRANS\ugtopv.exe"
set toRun=%toRun% @DB/BgStrukEx01/A
set toRun=%toRun% -pim=yes -u=%TceUser% -p=%TceUserPassword%

set JobParameter=-noudate
set JobParameter=%JobParameter% -config="%tessConfig_DPNE%"
set JobParameter=%JobParameter% -honour_structure

rem command vollständig zusammensetzen
set toRun=%toRun% %JobParameter%
goto :startJTGen
```





## Get Help

### NX JT Online Dokumentation

- `D:\ugs\ugnx5\UGDOC\html_files\ugtojt\index.html`

### Internet Verweise

- <http://www.jt2go.com/downloads/>
- <http://www.jtopen.com>  
ist eine Interessensgemeinschaft von Endanwendern, Hoftware-Herstellern, akademischen Partnern und unterstützenden Organisationen zur Förderung der JT-Technologie.

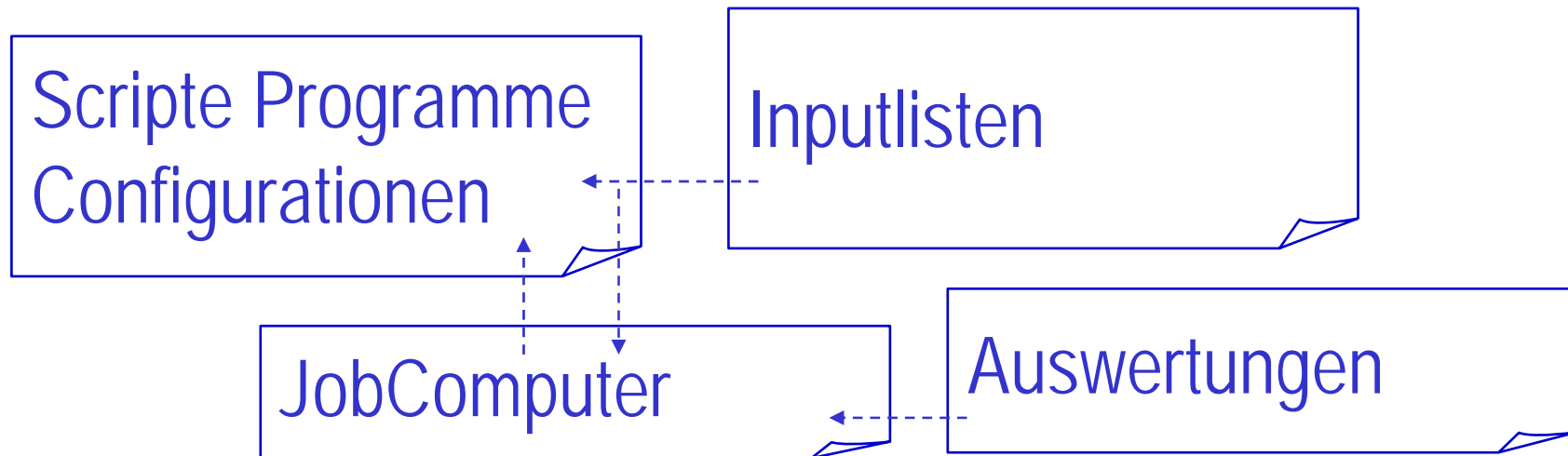


## Wie können solche Vorgänge insgesamt automatisiert werden

Es gibt im NX Umfeld verschiedenste Programme um Daten automatisch zu verarbeiten s0.z.B.

- NX Refile
- JT Erzeugung UgToPV
- DXF Konvertierungen via (ugto2d.exe → dxfdwg.exe)
- Parasolid Exoprts

Um dieser Programme einsetzen zu können benötigt man Scripte und die Liste der zu verarbeitenden Objekte. Eine Steuerung der Jobs auf mehreren Rechnern und eine Auswertung der Ergebnisse.



Im den folgenden Folien wird Beispielhaft gezeigt wie der Jobmanager eingesetzt werden kann um JT Files zu erzeugen.



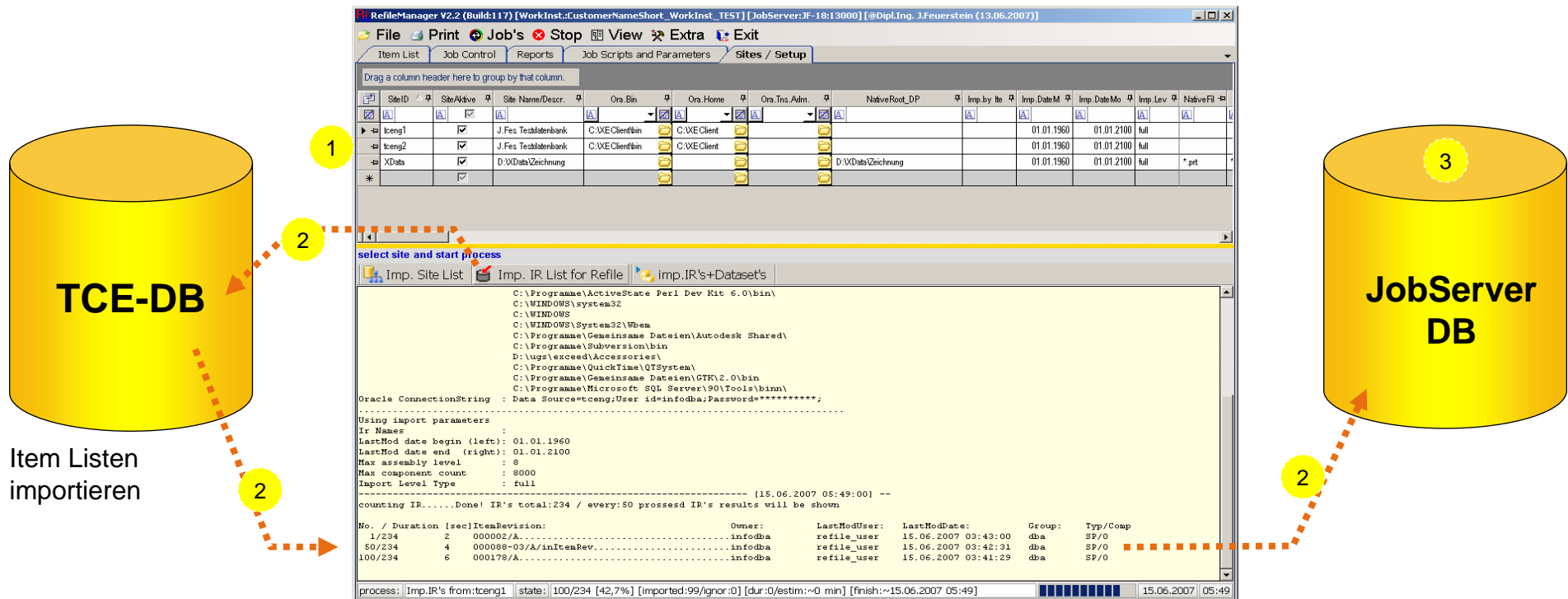
# Die Oberfläche des *PLMJobManager*

## Die Oberfläche unterteilt sich nach den Aufgabenbereichen

- Item List → Anzeige der Itemlisten und der Refile Ergebnissen
- Job Control → Organisation und Kontrolle der JobClnet Prozesse
- Reports → Auswertung der Job Ergebnisse (Listen Refile-Reports)
- Job Scripts and Parameters → Erfassung und Steuerung Job Parameter und Scripte
- Sites / Setup → Erfassung der Sites und Einlastung der Daten (Item's Datasets)

## PLMJobManager Site Setup und Vorbereitung der Datenbanken

Im JobServer werden die Sites der Unternehmen erfasst (1)  
 und die Item Listen in die JobServer-DB durch den Daten Import übertragen (2).  
 Diese Daten bilden die JobServer-DB (3) und sind die Datenbasis für die NX-Refile Prozesse.

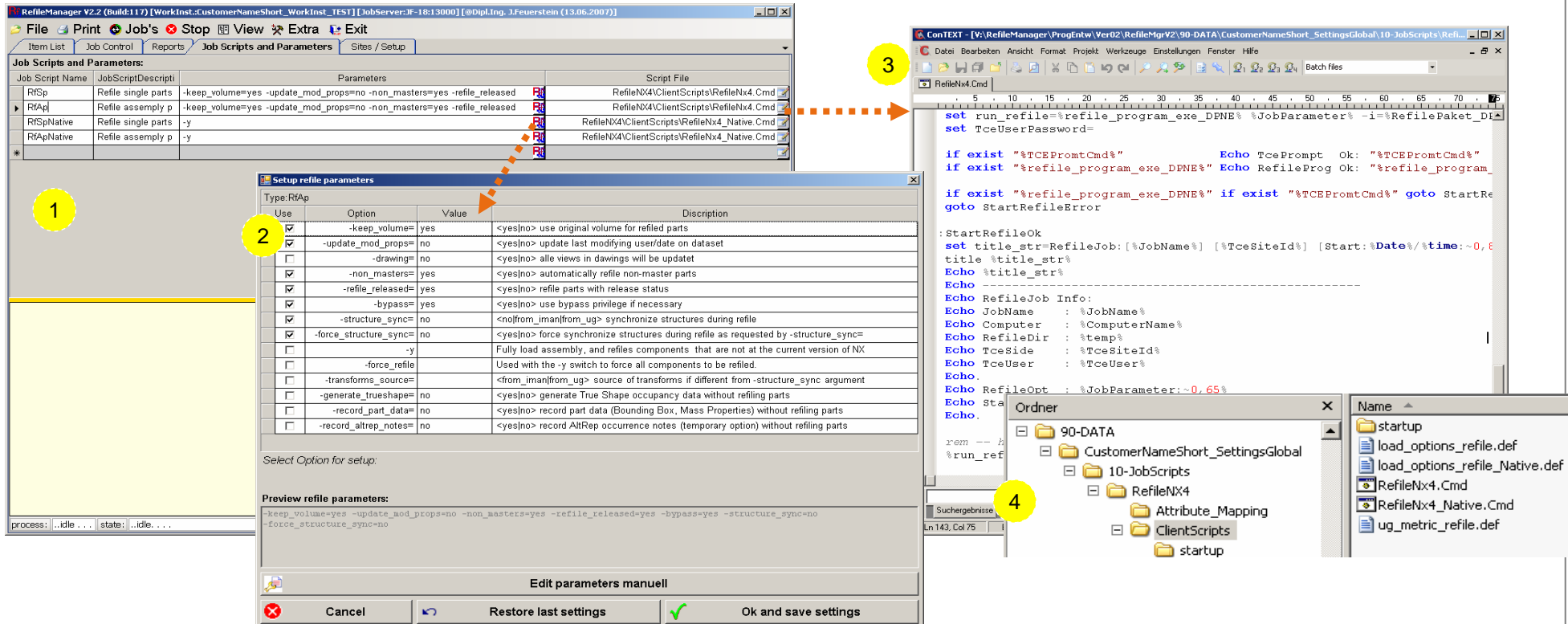


## PLMJobManager Steuerung der NX-Refile Optionen und Scripte

Im JobServer (1) wird die Steuerung der Parameter (2) vorgenommen. Zusätzlich werden die erforderlichen Scripte (3) und NX Einstellungs-Dateien (4) organisiert.

### Vorteil:

- ✓ Einfaches Verwalten der umfangreichen Parameter. (2)
- ✓ Übersichtliche Organisation der Scripte (3) und Einstelldateien (4)

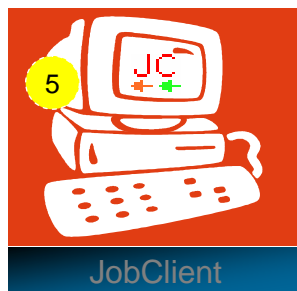


The screenshot illustrates the PLMJobManager interface for managing NX-Refile options and scripts. It is divided into several key areas:

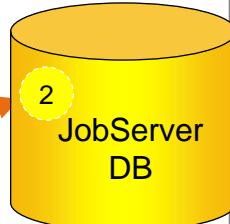
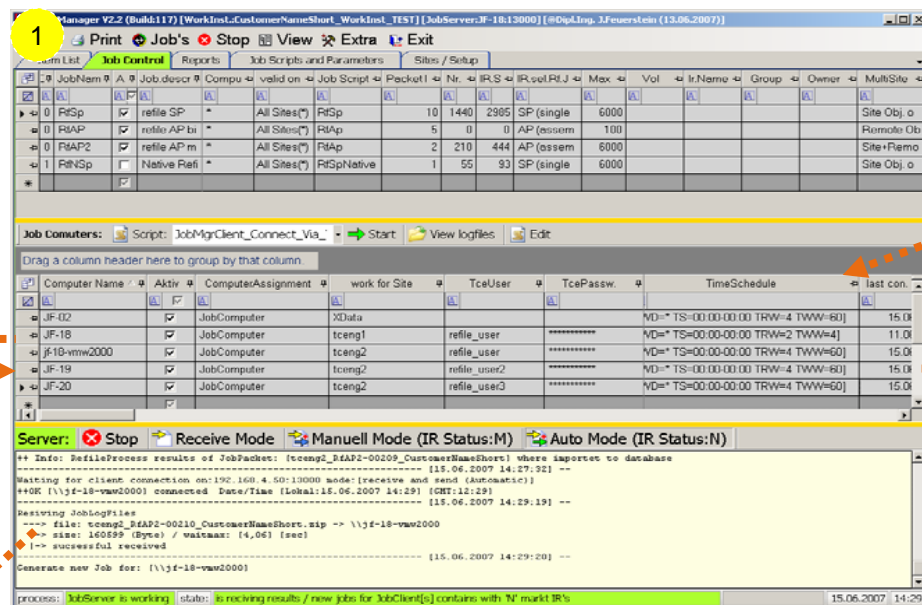
- Job Scripts and Parameters (1):** A table listing job scripts and their associated parameters. The table has columns for Job Script Name, JobScriptDescription, Parameters, and Script File.
- Setup refile parameters (2):** A dialog box for configuring refile parameters. It includes a table with columns for Use, Option, Value, and Discription. The parameters are organized into sections like 'Use', 'Option', 'Value', and 'Discription'.
- Script File (3):** A text editor showing the script file content, which includes commands for setting refile parameters and executing the refile process.
- File Explorer (4):** A file explorer showing the directory structure of the refile files, including folders like '90-DATA', 'CustomerNameShort\_SettingsGlobal', and '10-JobScripts'.

## Arbeiten mit dem JobServer Job Control

Der JobServer (1) wird zentral auf einem Arbeitsplatz installiert. Er organisiert die JobManager-DB (2) steuert alle JobProzesse (3) und und die Ablage aller Refile-Ergebnisse (4).

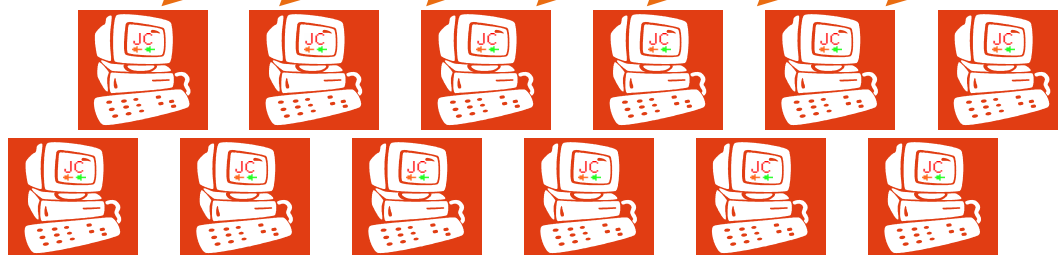
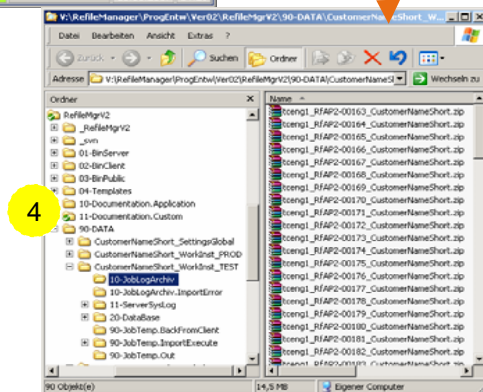


Ablauf Jobverarbeitung:  
 Die JobClients (5) verbinden sich mit dem JobServer (1) und rufen über diesen die Job-Prozesse ab (6). Der JobClient übermittelt seine Ergebnisse an den JobServer (7).



Der JobServer organisiert die empfangenen JobLogfiles

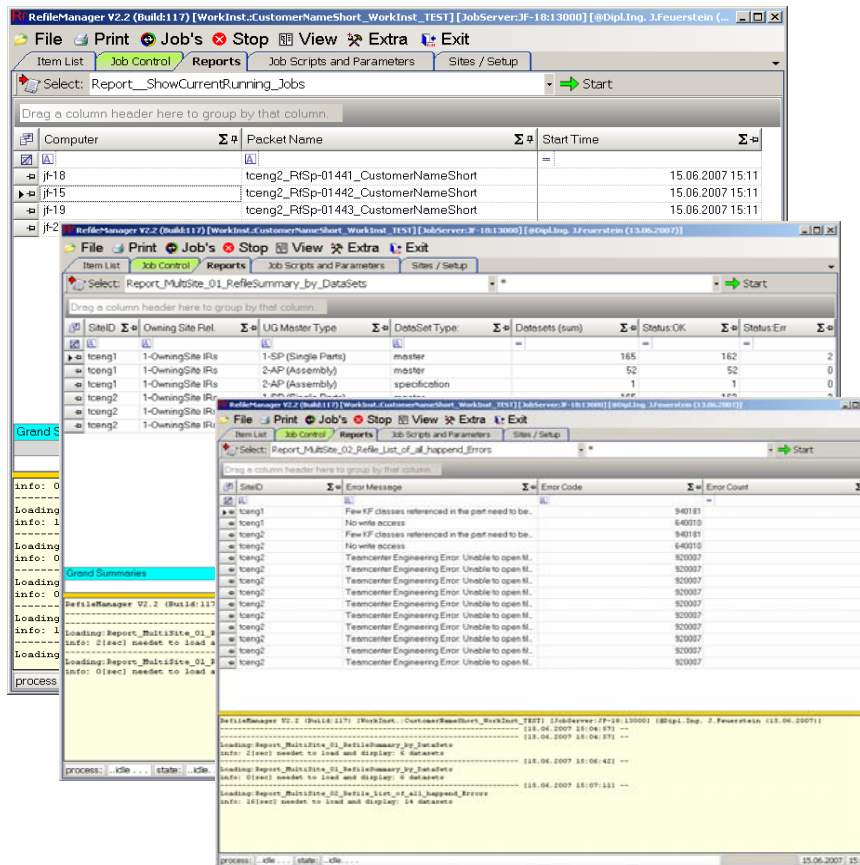
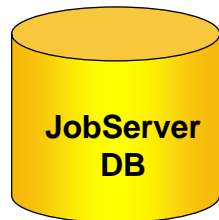
Koordinierung der JobClient's  
 = Job Steuerung



## Arbeiten mit dem JobServer: Reports

Das Reportsystem ermöglicht es:

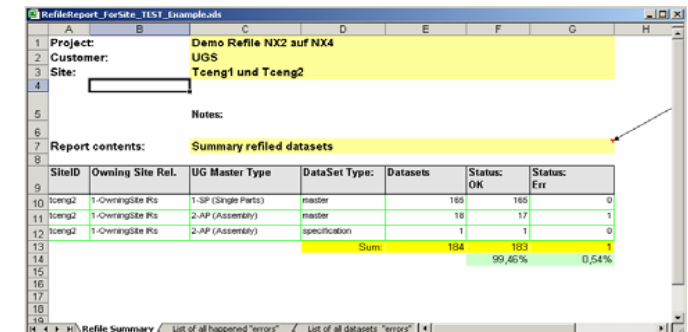
- die JobClnet Prozesse zu überwachen
- Job Ergebnisse auszuwerten (NX Ergebnisse).



Der Excel Export ermöglicht ein einfaches und schnelles ablegen aller Daten um z.B eigene Auswertungen oder Listen zu erstellen.



Über Copy können die Tabellen Daten in die Zwischenablage kopiert werden um diese z.B in einem Dokument wieder abzulegen



SiteID	Owning Site Rel.	UG Master Type	DataSet Type:	Datasets	Status: OK	Status: Err
10	1ceng2	1-OWningSite IRs	1-SP (Single Parts)	master	165	165
11	1ceng2	1-OWningSite IRs	2-AP (Assembly)	master	18	17
12	1ceng2	1-OWningSite IRs	2-AP (Assembly)	specification	1	1
Sum:				184	183	1
					99,45%	0,54%