

# PLMJobManager – NXPartUpdate

## Updating NX-Parts with Teamcenter environment

Author: Josef Feuerstein

# Table of content

|  |                |
|--|----------------|
| <a href="#"><u>Introduction</u></a>  | Slide: 3       |
| <a href="#"><u>Project Phase</u></a>   | Slide: 4       |
| <a href="#"><u>Chart Overview “Basic functionality” PLMJobManager</u></a>      | Slide: 5       |
| <a href="#"><u>Working with the JobServer: Job Control</u></a>                 | Slide: 6       |
| <a href="#"><u>Chart Overview “Multisite environment”</u></a>                  | Slide: 7       |
| <a href="#"><u>JobManager Database</u></a>                                     | Slide: 8       |
| <a href="#"><u>Setup and control of the PartUpdate options and scripts</u></a> | Slide: 9       |
| <a href="#"><u>Working with the JobServer: Reports</u></a>                     | Slide: 10      |
| <a href="#"><u>System Sketch TC + JobManager</u></a>                           | Slide: 11      |
| <a href="#"><u>How to PartUpdate?</u></a>                                      | Sides: 12 - 13 |
| <a href="#"><u>NX-PartUpdate with the PLMJobManager</u></a>                    | Slide: 14      |
| <a href="#"><u>Benefits Doing PartUpdate</u></a>                               | Slide: 15      |
| <a href="#"><u>Benefits of a NX-PartUpdate with the PLMJobManager:</u></a>     | Slide: 16      |
| <a href="#"><u>System requirements</u></a>                                     | Slide: 17      |

# Introduction

## What is PartUpdate ?

At any new version of NX, there are new functionality and modules. This causes changes of the data model.

When opening a Part-file, there is a check in which version it was saved.

When the saved version is older than the current one, NX is converting the data model inside the Part-file. – This is called PartUpdate

## Why PartUpdate?

From our development... we are able to open Part-files from Version 12.

Therefore we do not need a PartUpdate?

When working with Teamcenter, we do not have write access to loaded or PartUpdated parts.

Why do we have no write access?

- Part i owned by a different group
- Part is released
- Part has a different owning site

Parts from a prior NX-version get a modification flag after loading. If we PartUpdate the Parts during open, it will increase the loading time. Without write access, we do a PartUpdate every time, when opening a part or assembly.

Part is from an „old“ project ... does anybody know the NX-Version at that time? ... does anybody know that there are „old“ projects?

To have a clean NX-Environment, it is necessary to PartUpdate all Partfiles automatically and securely with the part\_utility.exe tool.

# Project Phase

## ■ **Phase 1 – Analysis**

- How many Datasets/Parts
- Analyse the environment (Precise/Imprecise, Loadoptions, Reference-Sets, Release status ...)
- Analyse and Setup the PartUpdate-Methods (order, settings and parameters)
- Check of disk space
- Clone the productive environment and Tests ...

## ■ **Phase 2 – Preparation and Test**

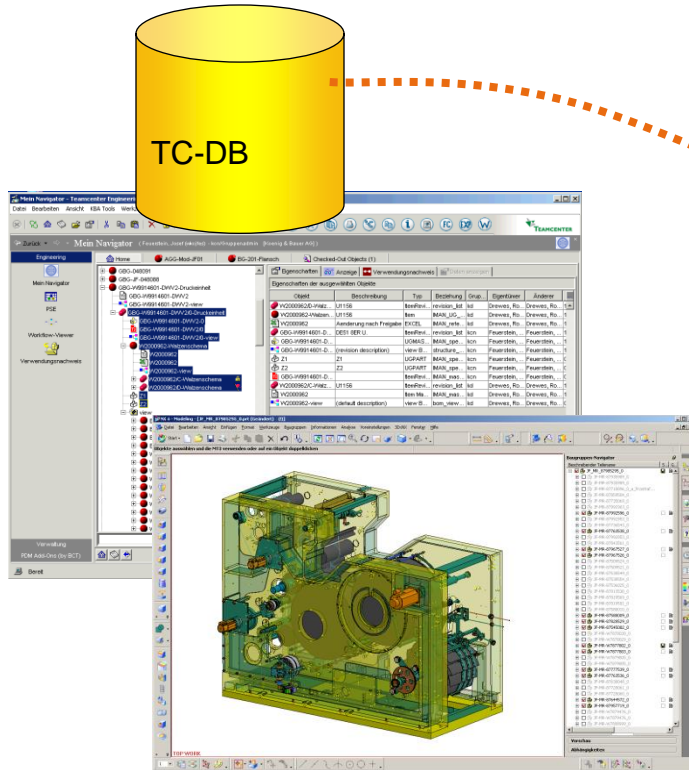
- PlmJobManager-OS-User and TC-User setup
- Check assemblies
- Access via Remote Desktop Connection or VNC
- Installation PlmJobManager -Manager-Tool (Server + Clients)
- Import PlmJobManager-Database from TCEng.
- Define Datapackage and Order for the PlmJobManager
- Define available Timeframes (day, night, take care of backup times)
- Check-Out Objects ... check them in, or mark them.
- Run PartUpdate-Tests
- Run CheckBox

## ■ **Phase 3 – Performing**

- Update the PlmJobManager-Database
- Explicit Check-Out Objects / User reference in a folder
- Check-Out Objects → prepare a Check In
- Run a Clearlocks
- Backup the Data
- PartUpdate-Packages at the PlmJobManager-Clients
- Analyse the PlmJobManager-Database
- Analyse the error PartUpdate-Parts
- Analyse the PartUpdate results ... create reports from the PlmJobManager

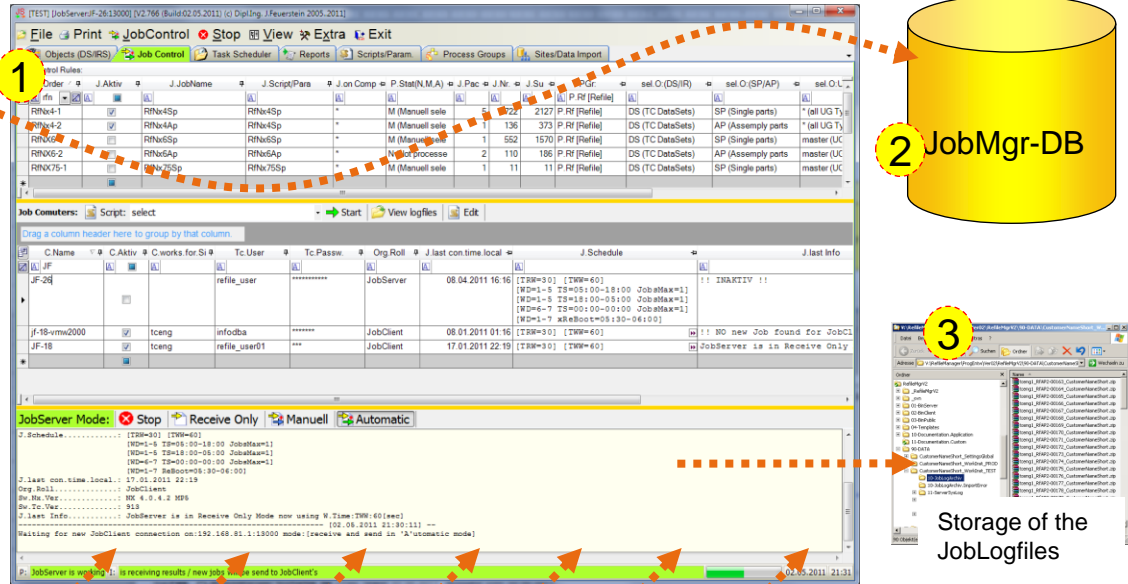
# Chart Overview “Basic functionality” PLMJobManager

## Teamcenter and NX environment

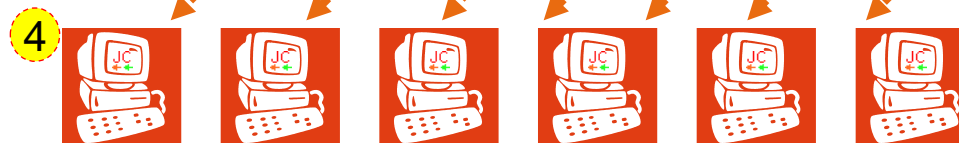


## PLMJobManager environment

All the jobs are administered by the JobServer. Furthermore, the JobServer controls the associated database (DB) (2) and manages the results of the jobs. The related log files are stored on a file server (3).



JobServer (1) Controls JobClients (4)

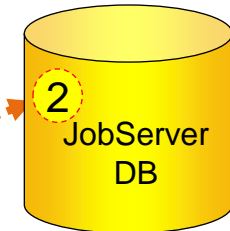
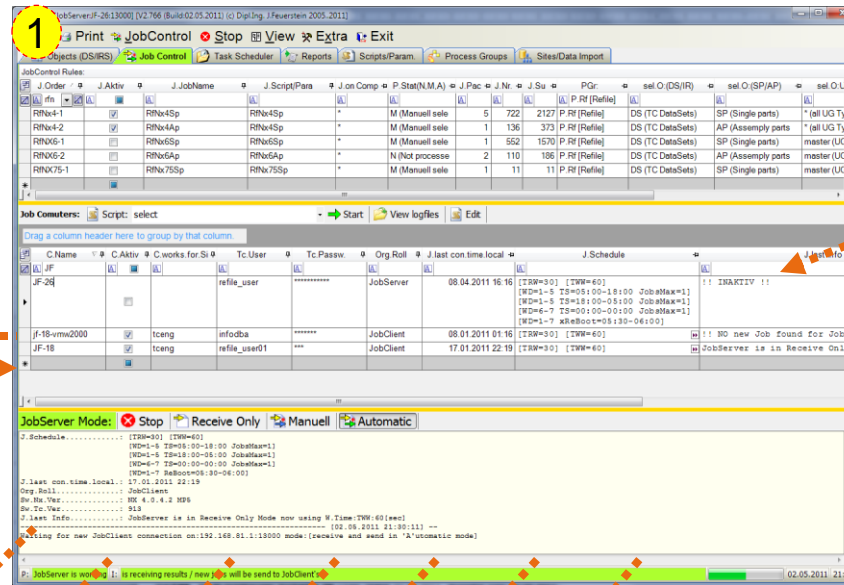


# Working with the JobServer: Job Control

The JobServer (1) will be located at one of your locations. It organizes the JobServer-DB (2) organizes all Job Clients (3) and storage of the PartUpdate results. (4)

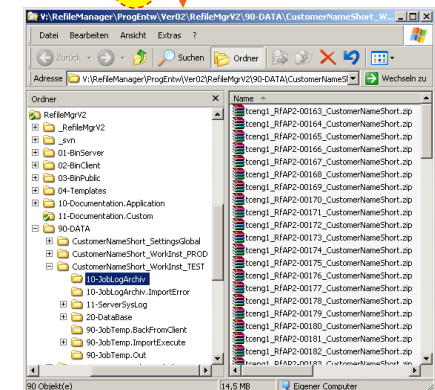
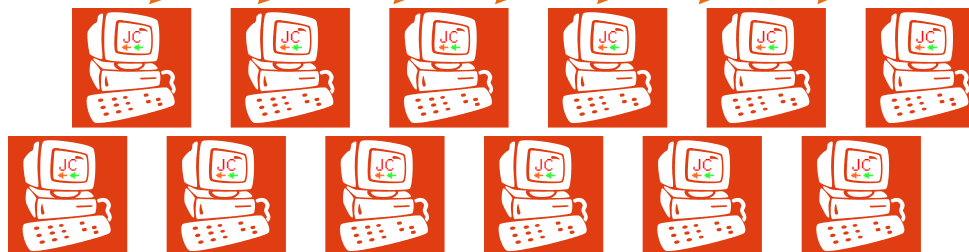


Procedure of the processing:  
The JobClients (5) are connecting to the JobServer (1) and receiving a PartUpdate Job (packet) (6). The JobClient hands over the Results to the JobServer (7).



The JobServer organizing the received JobLogfiles

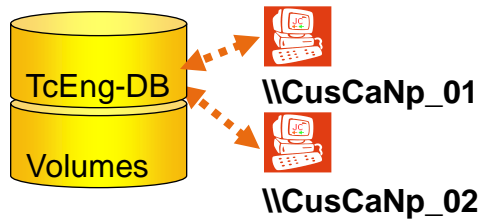
Coordination of the JobClient's



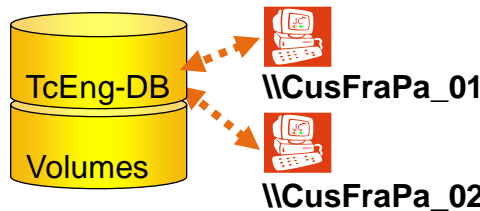
# Chart Overview “Multisite environment”

## Sites

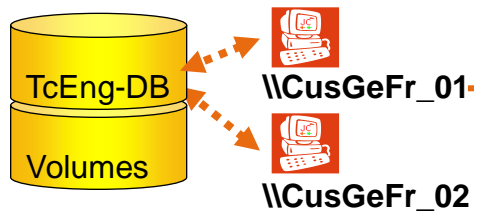
**Site: USA**  
**Location: Newport, California**



**Site: Europe1**  
**Location: Paris, France**



**Site: Europe2**  
**Location: Frankfurt, Germany**



## JobServer and NXPartUpdate environment

The JobServer will be located at one of your locations. From all sites the IR Lists for NX-PartUpdate will be imported to the JobServer Database

JobServer V2.31 (Build:09.02.2008) [WorkInst.:TEST] [JobServer:JF-18:13000] [©Dipl.Ing. J.Feuerstein]

File Print Job's Stop View Extra Exit

Items Rf. Job Control Reports Scripts/Param. Sites/Data Import

| Or | JobName         | Job description                         | Aktiv                               | valid on site | Job Script/Para | MultiSite Objects | Packet IRs si |
|----|-----------------|---|-------------------------------------|---------------|-----------------|-------------------|---------------|
| 01 | RfSpOwningParts | Refille Single Parts from Owning Site   | <input checked="" type="checkbox"/> | All Sites(*)  | RfSp            | S_Obj             |               |
| 02 | RfSpRemoteParts | Refille Single Parts from Remote Site   | <input checked="" type="checkbox"/> | All Sites(*)  | RfSp            | Remote Obj. only  |               |
| 03 | RfApOwningParts | Refille Assembly Parts from Owning Site | <input checked="" type="checkbox"/> | All Sites(*)  | RfAp            | S_Obj             |               |
| 04 | RfApRemoteParts | Refille Assembly Parts from Remote Site | <input checked="" type="checkbox"/> | All Sites(*)  | RfAp            | Remote Obj. only  |               |
| *  |                 |   | <input checked="" type="checkbox"/> |               |                 |                   |               |

Job Computers: Script: select Start View logfiles Edit

Drag a column header here to group by that column.

| Computer Na | Aktiv                               | work for Site | ComputerAssign | Schedule                               | Job.Info |
|-------------|-------------------------------------|---------------|----------------|--|----------|
| CusCaNp_01  | <input checked="" type="checkbox"/> | CusCaNP       | JobClient      | [TRW=4] [TWW=15] [WD=* TS=17:30-05:30] | aktive   |
| CusCaNp_02  | <input checked="" type="checkbox"/> | CusCaNP       | JobClient      | [TRW=4] [TWW=15] [WD=* TS=17:30-05:30] | aktive   |
| CusFraPa_01 | <input checked="" type="checkbox"/> | CusFraPa      | JobClient      | [TRW=4] [TWW=15] [WD=* TS=17:30-05:30] | aktive   |
| CusFraPa_02 | <input checked="" type="checkbox"/> | CusFraPa      | JobClient      | [TRW=4] [TWW=15] [WD=* TS=17:30-05:30] | aktive   |
| CusGeFr_01  | <input checked="" type="checkbox"/> | CusGeFr       | JobClient      | [TRW=4] [TWW=15] [WD=* TS=17:30-05:30] | aktive   |
| CusGeFr_02  | <input checked="" type="checkbox"/> | CusGeFr       | JobClient      | [TRW=4] [TWW=15] [WD=* TS=17:30-05:30] | aktive   |
| *           | <input checked="" type="checkbox"/> |               |                |  |          |

Set Server Mode Stop Receive Only Obj.with Status:M (Manuell) Obj.with Status:N (Not Pro

```

SiteId      : CusCaNP
Node Name   : CusCaNP_01 (Mem:1023Mb)
Node NX Version : v4.0.3.3
Node TCE Version : 913
RefilleJob.Log : _RefilleJob.LOG
Refille IR list : CusCaNP_RfSp-00006_CustomerNameShort_ObjectsList.txt
Refille comands : -keep_volume=yes -update_mod_props=no -non_masters=yes -refile_released=yes -bypass=yes

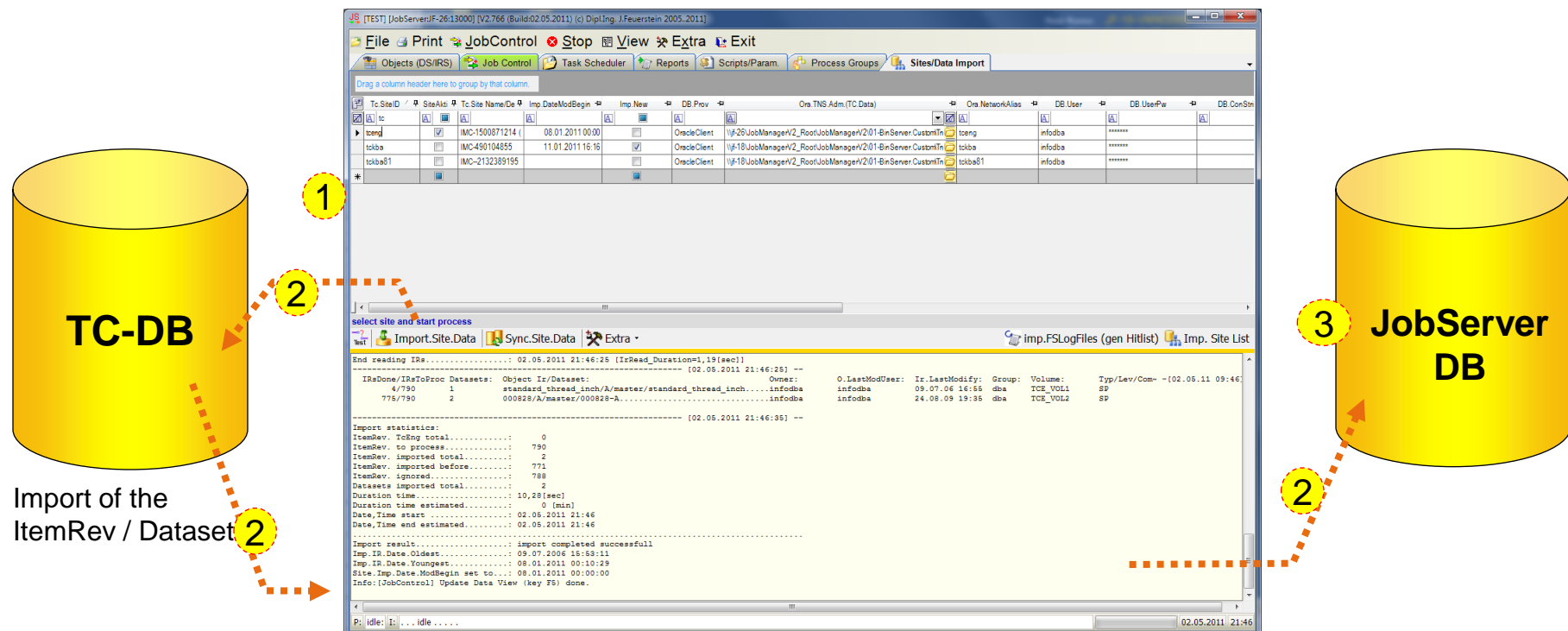
Site:      ItemNameRev:      DataSetName:      DataSetType:      ErrCode:      Process Time Start:
CusCaNP    000128/A          000128/A          master           0             18.02.2008 14:12:43

process: ..idle ... state: ..idle ...
    
```

# JobManager Database

## Load the data into the PLMJobManager

- In the JobServer we setup all customer Sites (1)
- Import all ItemRev and Datasets into the JobServer-DB (2).
- This is the content of the JobServer-DB (3) and they are the database for the Job Process



# Setup and control of the PartUpdate options and scripts

The setup and control of the NX-PartUpdate Parameter (2) is done inside the JobServer (1). Also the needed scripts (3) and the NX-PartUpdate setup (4) will be organized there.

## Benefit:

- ✓ Easy administration of all NX-PartUpdate parameters. (2)
- ✓ Clear organization of the scripts (3) and setups (4)

The screenshot illustrates the setup and control of the NX-PartUpdate options and scripts. It shows the RefileManager V2.2 interface with the Job Scripts and Parameters tab selected. The Job Scripts and Parameters table lists various scripts and their parameters. The Setup refile parameters dialog is open, showing the configuration for the RTAp script. The file structure of the RefileNX4 scripts is also shown, organized into folders like 90-DATA, CustomerNameShort\_SettingsGlobal, 10-JobScripts, and RefileNX4.

**1** Job Scripts and Parameters:

| Job Script Name | JobScriptDescription | Parameters  | Script File                                  |
|-----------------|----------------------|---|--|
| RTSp            | Refile single parts  | -keep_volume=yes -update_mod_props=no -non_masters=yes -refile_released | RefileNX4\ClientScripts\RefileNx4.Cmd        |
| RTAp            | Refile assembly p    | -keep_volume=yes -update_mod_props=no -non_masters=yes -refile_released | RefileNX4\ClientScripts\RefileNx4.Cmd        |
| RTSpNative      | Refile single parts  | -y  | RefileNX4\ClientScripts\RefileNx4_Native.Cmd |
| RTApNative      | Refile assembly p    | -y  | RefileNX4\ClientScripts\RefileNx4_Native.Cmd |

**2** Setup refile parameters:

| Use                                 | Option                 | Value | Description   |
|-------------------------------------|------------------------|-------|---|
| <input checked="" type="checkbox"/> | -keep_volume=          | yes   | <yes no> use original volume for refiled parts  |
| <input checked="" type="checkbox"/> | -update_mod_props=     | no    | <yes no> update last modifying user/date on dataset                                   |
| <input type="checkbox"/>            | -drawing=              | no    | <yes no> alle views in drawings will be updated                                       |
| <input checked="" type="checkbox"/> | -non_masters=          | yes   | <yes no> automatically refile non-master parts  |
| <input checked="" type="checkbox"/> | -refile_released=      | yes   | <yes no> refile parts with release status   |
| <input checked="" type="checkbox"/> | -bypass=               | yes   | <yes no> use bypass privilege if necessary  |
| <input checked="" type="checkbox"/> | -structure_sync=       | no    | <no from_iman from_ug> synchronize structures during refile                           |
| <input checked="" type="checkbox"/> | -force_structure_sync= | no    | <yes no> force synchronize structures during refile as requested by -structure_sync=  |
| <input type="checkbox"/>            | -y                     |       | Fully load assembly, and refiles components that are not at the current version of NX |
| <input type="checkbox"/>            | -force_refile          |       | Used with the -y switch to force all components to be refiled.                        |
| <input type="checkbox"/>            | -transforms_source=    |       | <from_iman from_ug> source of transforms if different from -structure_sync argument   |
| <input type="checkbox"/>            | -generate_trueshape=   | no    | <yes no> generate True Shape occupancy data without refileing parts                   |
| <input type="checkbox"/>            | -record_part_data=     | no    | <yes no> record part data (Bounding Box, Mass Properties) without refileing parts     |
| <input type="checkbox"/>            | -record_altrp_notes=   | no    | <yes no> record AltRep occurrence notes (temporary option) without refileing parts    |

Select Option for setup:

Preview refile parameters:

```

-keep_volume=yes -update_mod_props=no -non_masters=yes -refile_released=yes -bypass=yes -structure_sync=no
-force_structure_sync=no
  
```

**3** RefileNX4.Cmd:

```

set run_refile=%refile_program_exe_DPNE% %JobParameter% -i=%RefilePaket_Di
set TceUserPassword=

if exist "%TCEPromtCmd%" Echo TcePromt Ok: "%TCEPromtCmd%"
if exist "%refile_program_exe_DPNE%" Echo RefileProg Ok: "%refile_program_

if exist "%refile_program_exe_DPNE%" if exist "%TCEPromtCmd%" goto StartRe
goto StartRefileError

:StartRefileOk
set title_str=RefileJob: [%JobName%] [%TceSiteId%] [Start: %Date%/%time:~0,6
title %title_str%
Echo %title_str%
Echo -----
Echo RefileJob Info:
Echo JobName : %JobName%
Echo Computer : %ComputerName%
Echo RefileDir : %temp%
Echo TceSide : %TceSiteId%
Echo TceUser : %TceUser%
Echo.
Echo RefileOpt : %JobParameter:~0,65%
Echo Sta
Echo.
  
```

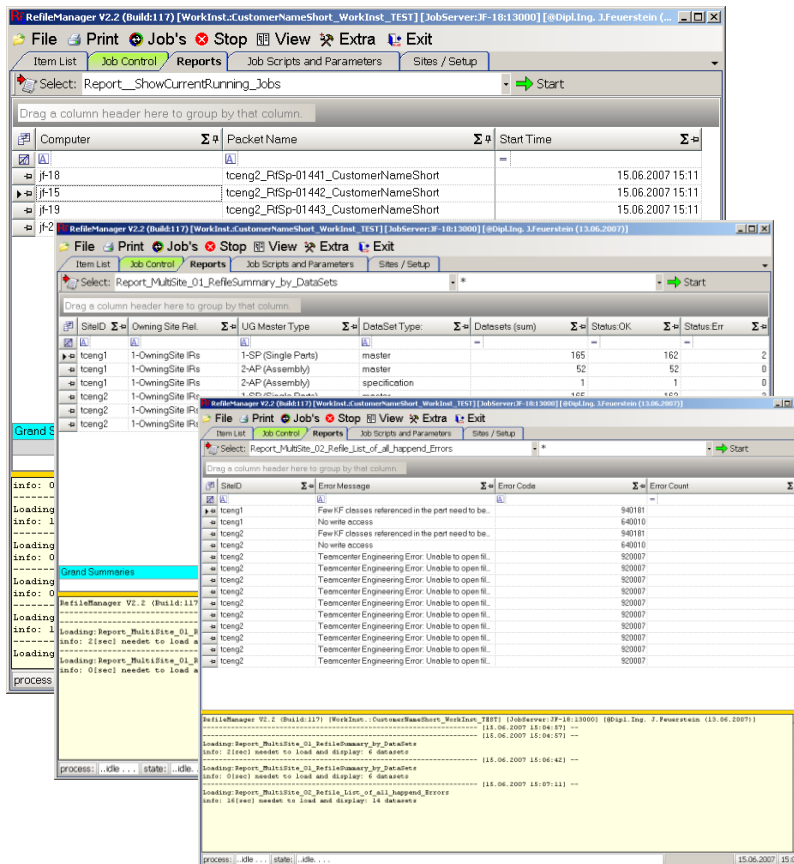
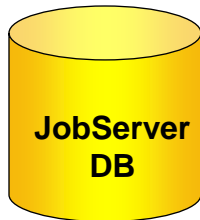
**4** File structure:

- 90-DATA
  - CustomerNameShort\_SettingsGlobal
    - 10-JobScripts
      - RefileNX4
        - Attribute\_Mapping
        - ClientScripts
          - startup

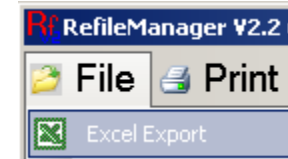
# Working with the JobServer: Reports

## Reporting functionality:

- Monitoring of the JobClient processes
- Analysis of the PartUpdate-Logfiles



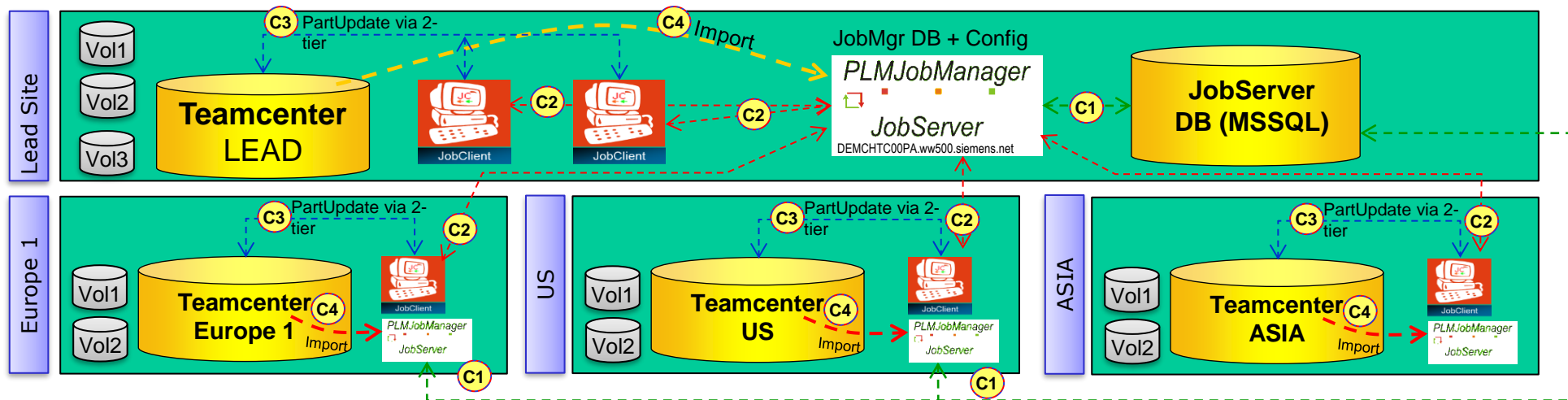
The **Excel Export** functionality allows you to easily store all needed data for further own calculations.



Use the clipboard (copy/paste) functionality if some data is needed in your own documents.

| SiteID | Owning Site Rel. | UG Master Type    | DataSet Type:       | Datasets | Status: OK | Status: Err |
|--------|------------------|-------------------|---------------------|----------|------------|-------------|
| 10     | 1ceng2           | 1-OwningSite (Rs) | 1-SP (Single Parts) | 165      | 165        | 0           |
| 11     | 1ceng2           | 1-OwningSite (Rs) | 2-AP (Assembly)     | 18       | 17         | 1           |
| 12     | 1ceng2           | 1-OwningSite (Rs) | 2-AP (Assembly)     | 1        | 1          | 0           |
| 13     |                  |                   | Sum                 | 184      | 183        | 1           |
| 14     |                  |                   |                     |          | 99,46%     | 0,54%       |

# System Sketch TC + JobManager



| Nr | Description                         | Connect via:                |
|----|-------------------------------------|-----------------------------|
| C1 | communication JobServer MS-SQL      | TCP:1433 UDP: 1434          |
| C2 | communication JobClient – JobServer | Port:13000 / 13001          |
| C3 | part_utility.exe (2-tier)           | Handel by IT                |
| C4 | Import Meta Data from TC Db         | Uses TnsNames.ora like (C3) |

## System prerequisite

1. Teamcenter inc. all Volumes Data
2. The Volumes need to have +1/3 free disk space  
Example.: fpr 100 GB NX Data we need 33 GB of free disc space
3. Oracle Read Only User reading data from TC. Database.
4. TC PartUpdate User with DBA right
5. TC PartUpdate User mast have read write right's on all Volumes
6. Script to get TC Prompt.
7. JobClients with TC 2Tier client and NX in the correct version
8. Remote Access to NX PartUpdate JobClient's
9. 600 MB of Network disk space for PLMJobmanager Software Installation and configuration
10. ~1 GB Network diskpace for JobProcess Logfiles for each 250.000 Parts to PartUpdate

# How to PartUpdate?

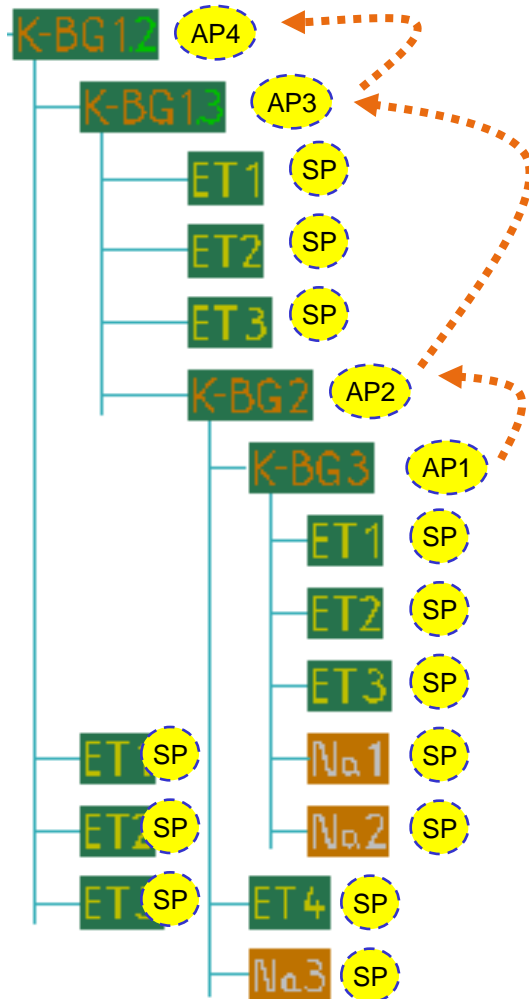
## For a optimum PartUpdate Result we defined the following Steps for convert mating conditions:

- Step 1: Pu. Last Used Singelpart's
- Step 2: Pu. Last Used Assembly Part`s → Base PartUpdate
- Step 3: Pu. Last Used Assembly Part`s → -convert\_mcs
- Step 4: Pu. Remaining Singelpart's
- Step 5: Pu. Remaining Assembly Part`s → Base PartUpdate
- Step 6: Pu. Remaining Assembly Part`s → -convert\_mcs
- Step 7: Generate Result Report

**During PartUpdate Process and specially between the PartUpdate Steps it is recommended to Analyze the PartUpdate Results.**

# How to Update you NX-Parts?

Bottom - Up Prinzip:



**The Part-Update of the Nx Data should be done via Bottom-Up Principe.**

**The advance is that the Update of Assembly is systematic processed from Bottom Up. It reduces the update time and improves the final Data quality**

**Therefore the Parts and Processed in this order:**

- First all = Single Part's (SP)
- All Assembly's (AP) started with the lowest mound of levels and components.

|     |               |              |
|-----|---------------|--------------|
| AP1 | has 1 Level   | 5 Component  |
| AP2 | has 2 Level's | 8 Component  |
| AP3 | has 3 Level's | 12 Component |
| AP4 | has 4 Level's | 16 Component |

# NX-PartUpdate with the *PLMJobManager*

It supports and optimizes the upgrade to a higher NX-Version within a unique procedural method.

## **The following methods are supported from the PLMJobManager:**

- ✓ processing sequence of the NX Partfiles:
  - Single Parts → recent parts at first, older parts at last
  - Assemblies → with the „Button Up“ principle  
(Button Up = dependent of the amount of components, from button to top)
- ✓ Adjustments for processing sequence i.e. volumes, groups, projects etc...
- ✓ Analysis of the PartUpdate-Logfiles for Status, error messages, warnings, etc..
- ✓ Storage of the Client NX-PartUpdate-Logfiles.
- ✓ Central control of the PartUpdate processes in a Multisite environment

# Benefits Doing PartUpdate

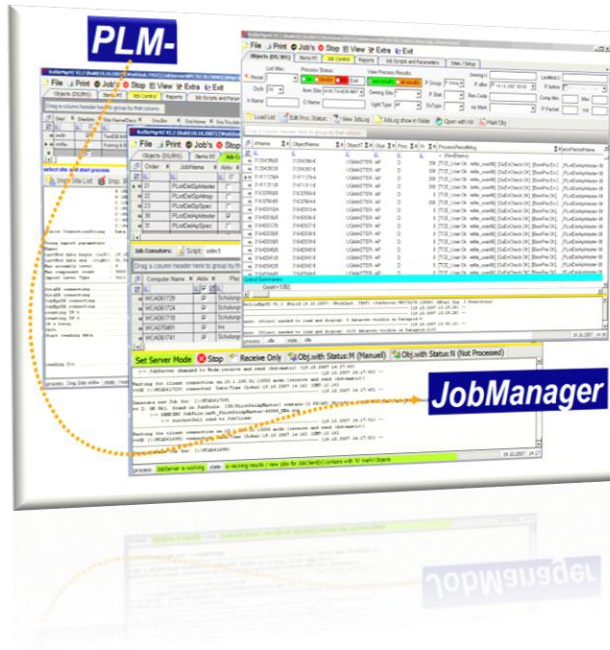
- ✓ After PartUpdate all NX Data are saved in new NX Version  
→ Designers working with homogeneous NX Dataversions
- ✓ Released and Partfamily Members will be saved in new Version  
→ Designers cannot save them interactive.
- ✓ NX Multisite - Replica Data will also be saved in new Version  
→ Designers cannot save them interactive.
- ✓ With PartUpdate you can convert Mating Conditions to assembly constraints.  
→ Less work for designers
- ✓ PartUpdate creates all lightweight representations  
→ all parts have then lightweight representations
- ✓ Improvement on loading Parts and Assembly's  
→ because no conversion is needed during load process.

# Benefits of a NX-PartUpdate with the PLMJobManager:

The PLMJobManager organizes the conversion of the NX-Data with the NX-PartUpdate utility inside a Teamcenter environment.

- ✓ Unique setup and control of the PartUpdate-Options and Processes.
- ✓ Distribution of the PartUpdate process to multiple computer-clients. (Less PartUpdate process time)
- ✓ Running multiple processes on PartUpdate Client (max. usage of a Client)
- ✓ Avoid „multiple“ PartUpdate of NX-Parts.
- ✓ TC Single and Multisite is supported
- ✓ Process Monitoring Result validation and logging of PartUpdate processes
- ✓ Easy Result Reporting is integrated in JobManager.

# System requirements



## JobServer:

- Win10/Win11 64 bit
- WinServer 2008 - 2016 64 bit
- SQL Database for JobServer Application



## JobClient:

- Win10/Win11 64 bit
- WinServer 2008 - 2019 64 bit
- with full NX-2tier and TC installation

## Contact

Tel.: : +49 6682 – 9191-01

Email : [Josef.Feuerstein@addPLM.com](mailto:Josef.Feuerstein@addPLM.com)

Web : [www.addPLM.com](http://www.addPLM.com)