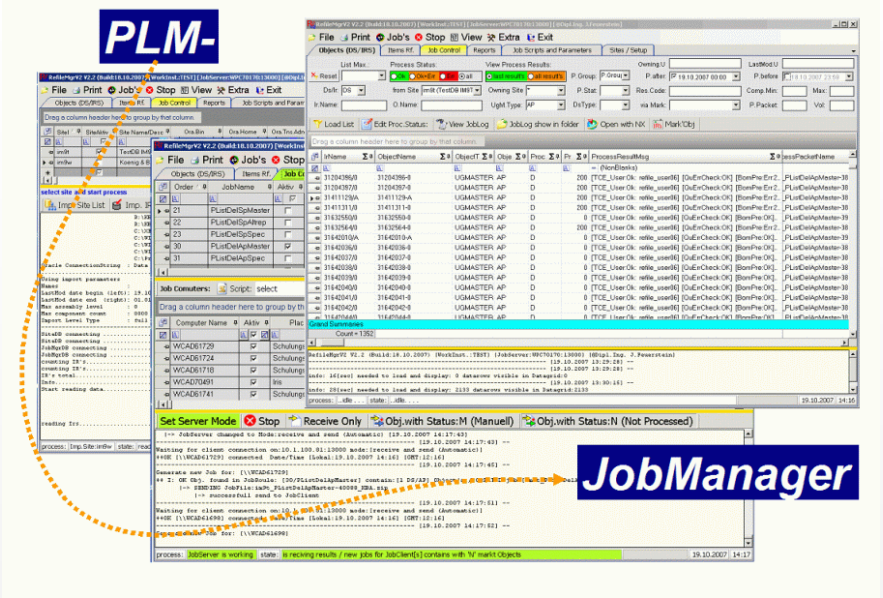
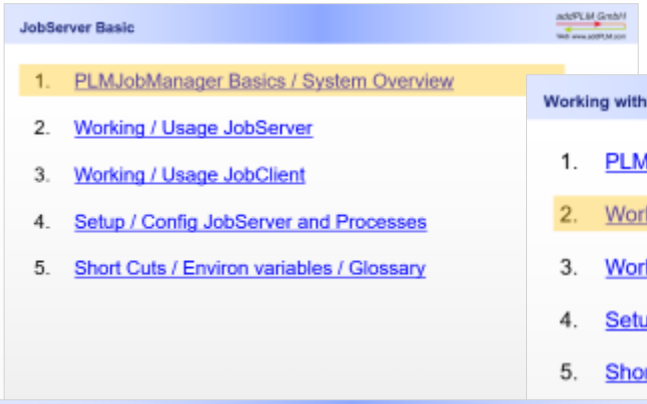


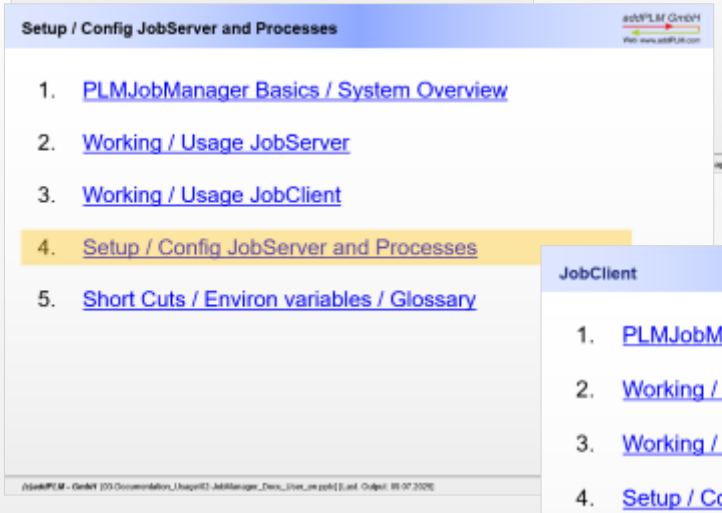
PLMJobManager

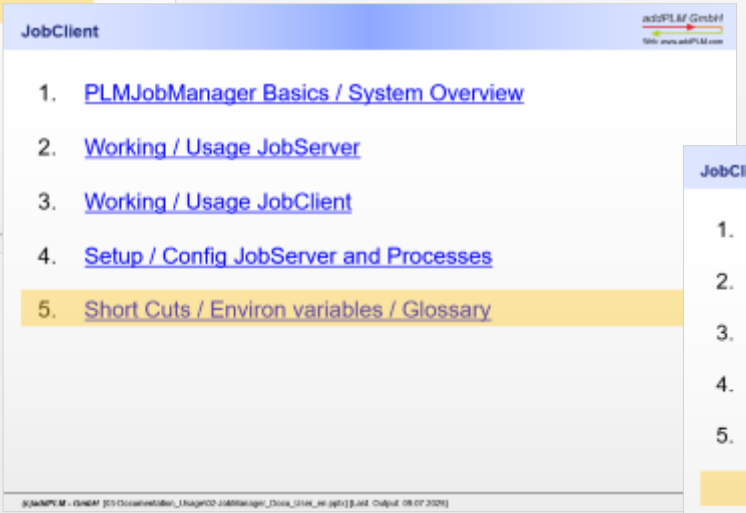
Usage and Configuration Documentation

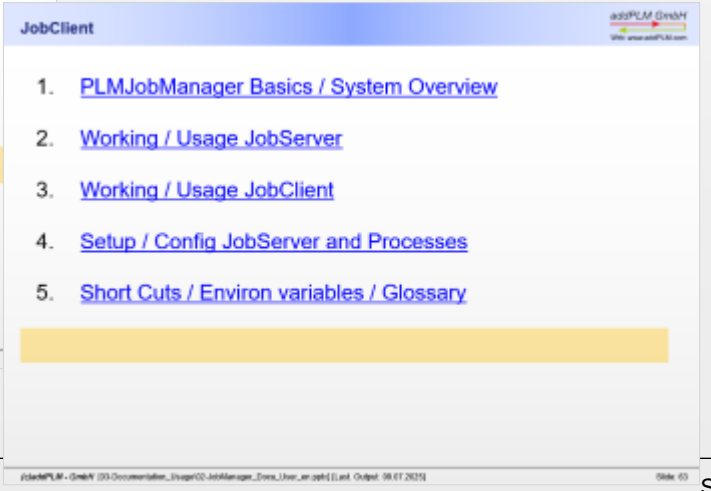
Created by : Josef Feuerstein

Overview - Content









PLMJobManager Basics

- How to process PLMData?

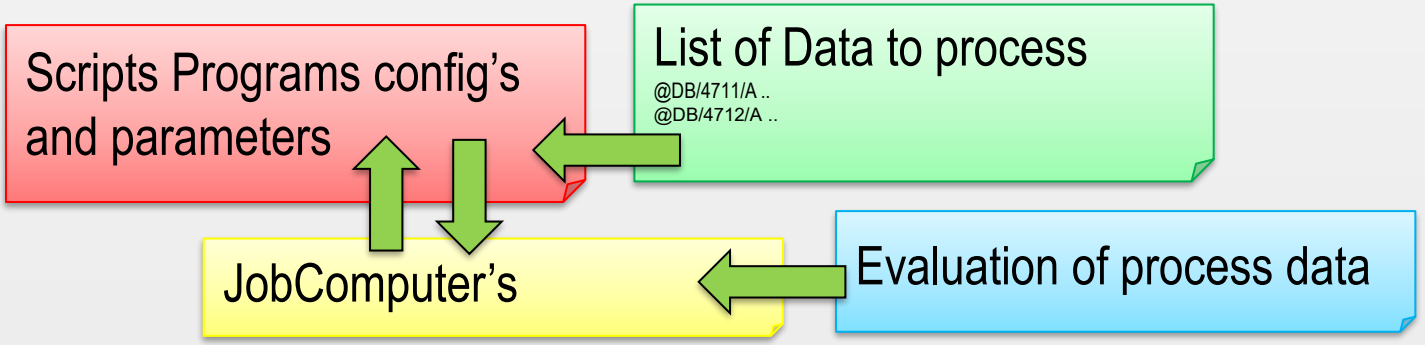
In TC and NX Environment there are many tools to process data in Batch mode like:

- Data migration via → TCIn_Import.exe
- Extracting NX Data information's via → CheckBox for NX
- JT-File generation via → UgToPV
- DXF-File Export via → ugto2d.exe und dxfdwg.exe
- IGES and Step Export
- CMM migration ProE/Catia to NX ..

To run those kind of program's we do need:

- Scripts configurations files and parameters to control a process under defined conditions
- Input list organized in right order of objects to process
- To process them on many computer in the same time.
- also it is needed to have a result evaluation of the process .

Schema:



The Concept of ***PLMJobManager*** is: to automate this kind of processes in a performant and understandable way.

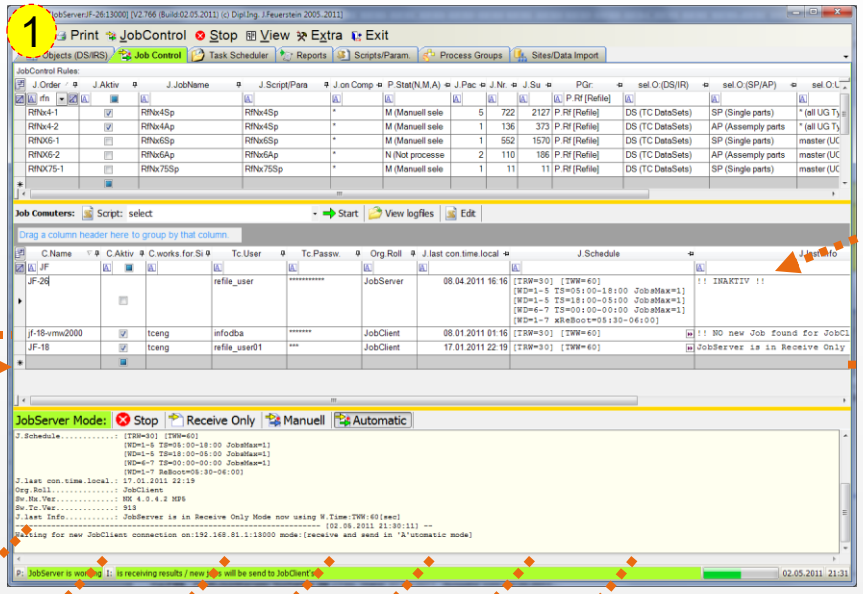
PLMJobManager Basics

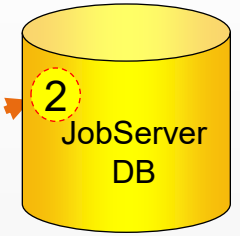
- PLMJobManager - Process Sketch

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

6

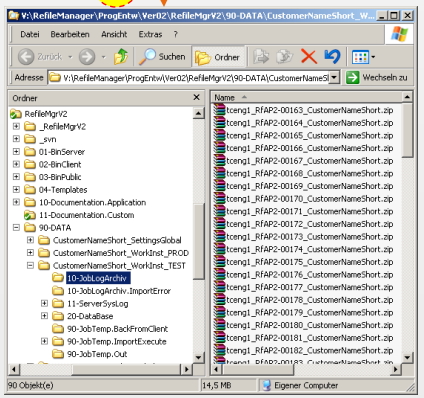
7



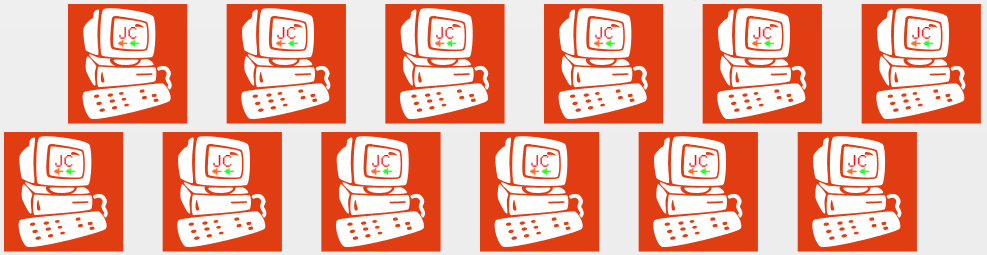


The JobServer organizing the received JobLogfiles

4



3 Coordination of the JobClient's

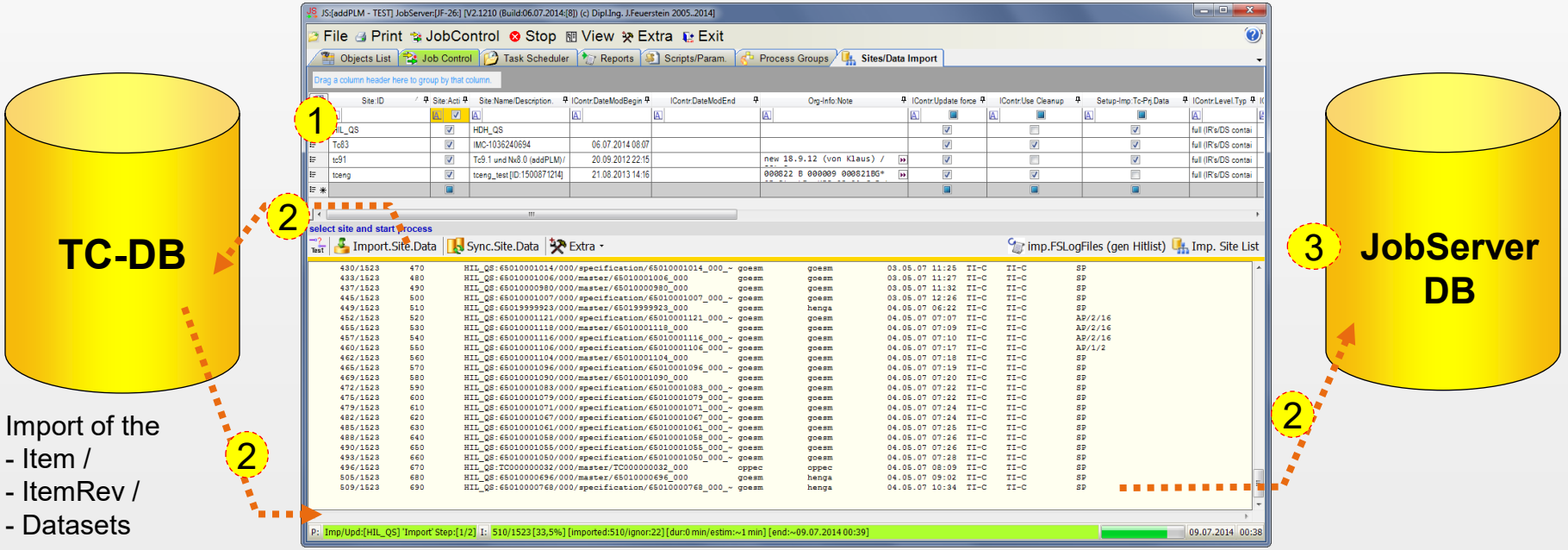


Procedure of the processing:
The JobClients **(5)** are connecting to the JobServer **(1)** and receiving a Job (packet) **(6)**. The JobClient hands over the Results to the JobServer **(7)**.

- JobManager Database

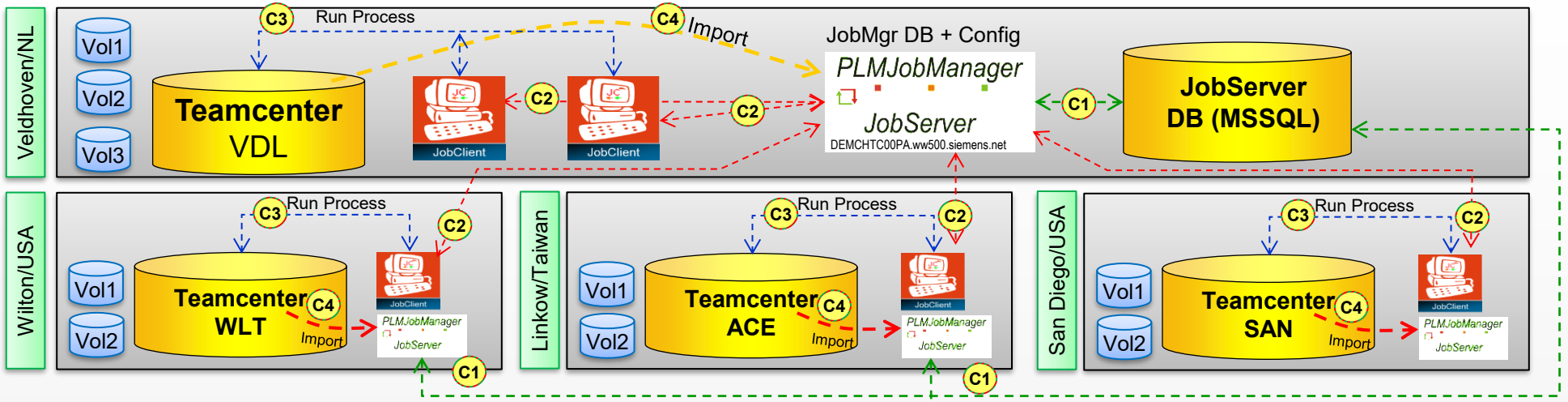
Load the data into the PLMJobManager

- In the PLMJobServer we register 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



PLMJobManager Basics

System Sketch TC + JobManager



Nr	Description	Connect via:
C1	communication JobServer MS-SQL	TCP:1433 UDP: 1434
C2	communication JobClinet – JobServer	Port:13000 / 13001
C3	TC/NX.exe process application	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 sufficient free diskspace
3. Oracle Read Only User reading data from TC. Database.
4. TC User with DBA rights or sufficient rights to do the process
5. TC User mast have sufficient rights to do the process on Volumes
6. Script to get TC Prompt.
7. JobClients with TC + NX installation in the correct version
8. Remote Access to JobClient‘s
9. 1 GB of Network disk space for PLMJobmanager Software Installation and configuration
10. ~1 GB Network diskspace for JobProcess Logfiles for each 250.000 Parts to process.

- Chart Overview “Multisite environment”

Sites

Site: USA
Location: Newport, Carlifornia

Tc-DB
Volumes
\\CusCaNp_01
\\CusCaNp_02

Site: Europe1
Location: Paris, France

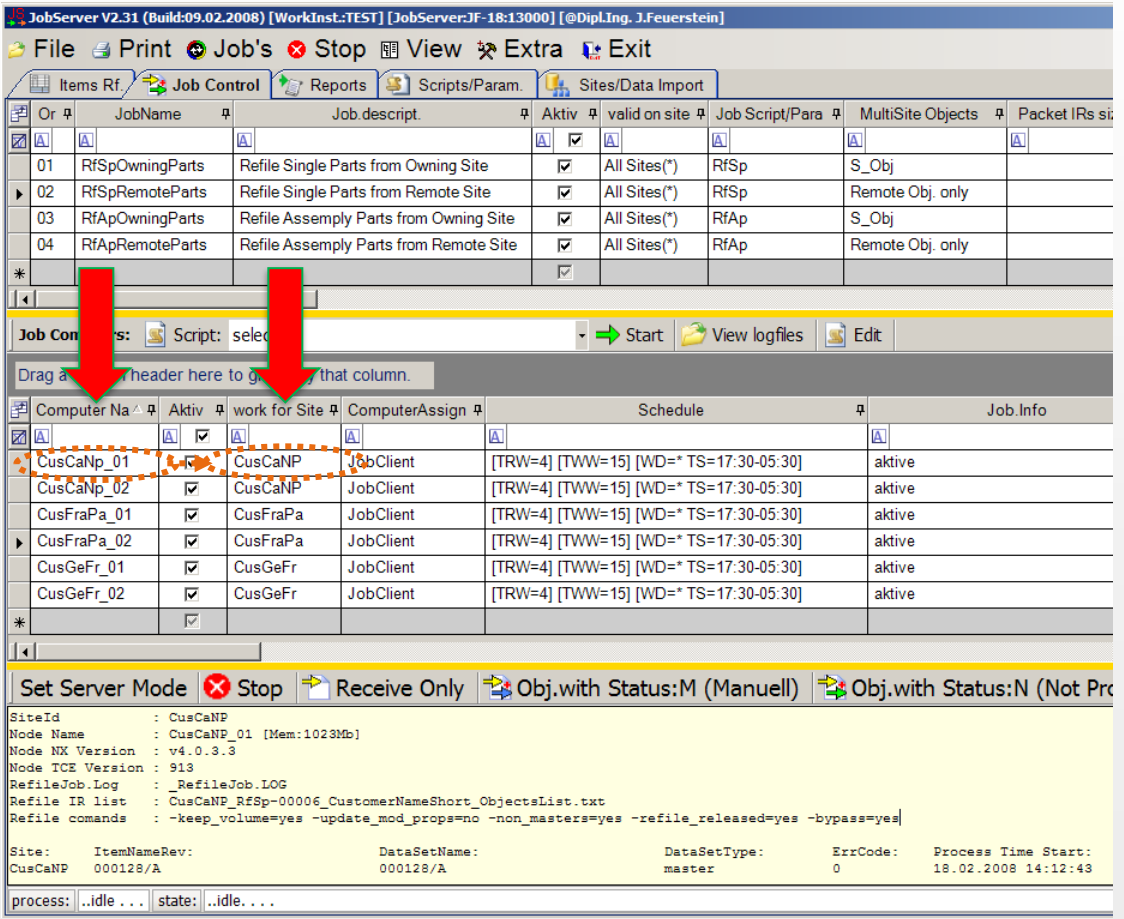
Tc-DB
Volumes
\\CusFraPa_01
\\CusFraPa_02

Site: Europe2
Location: Frankfurt, Germany

Tc-DB
Volumes
\\CusGeFr_01
\\CusGeFr_02

JobServer in Multisite Environment

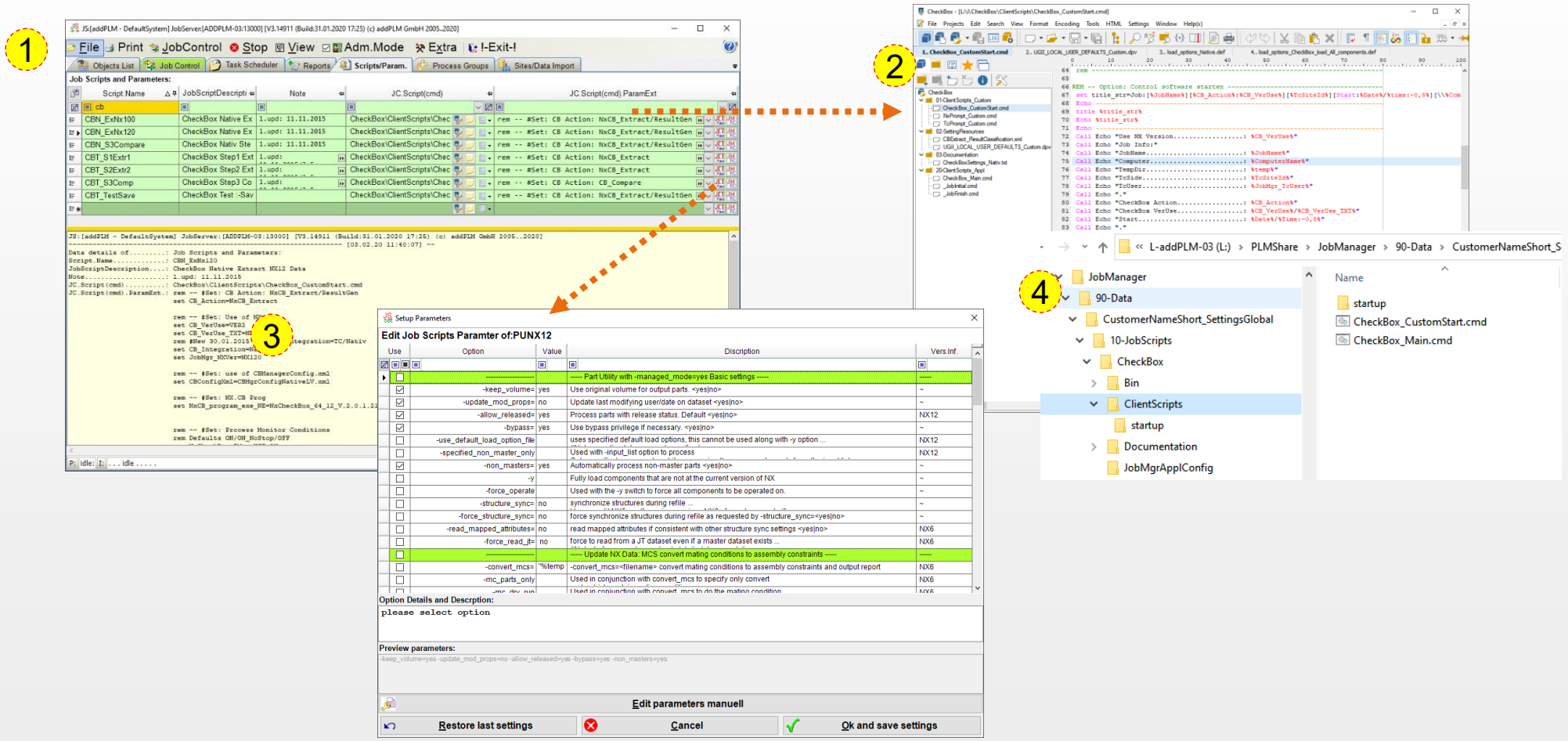
The JobServer will be located at one of your locations. From all sites the JobClients will get the Jobs to process on his Site.



PLMJobManager Basics

- Part 1 of configuration JobServers

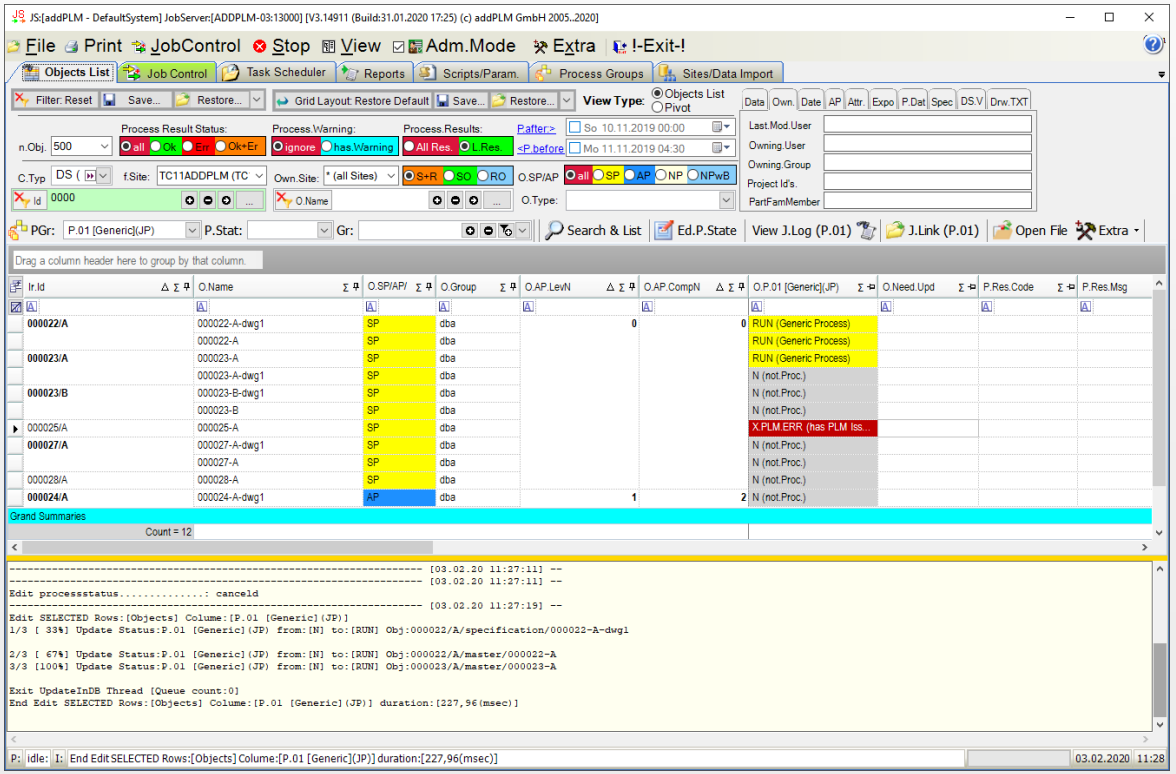
in JobServer **(1)** definition of scripts **(2)** Parameter **(3)** and script files **(4)** defines the base of a Job.
This definitions will be used in JobControl.



PLMJobManager Basics

- The GUI of PLMJobServer is organized in following section

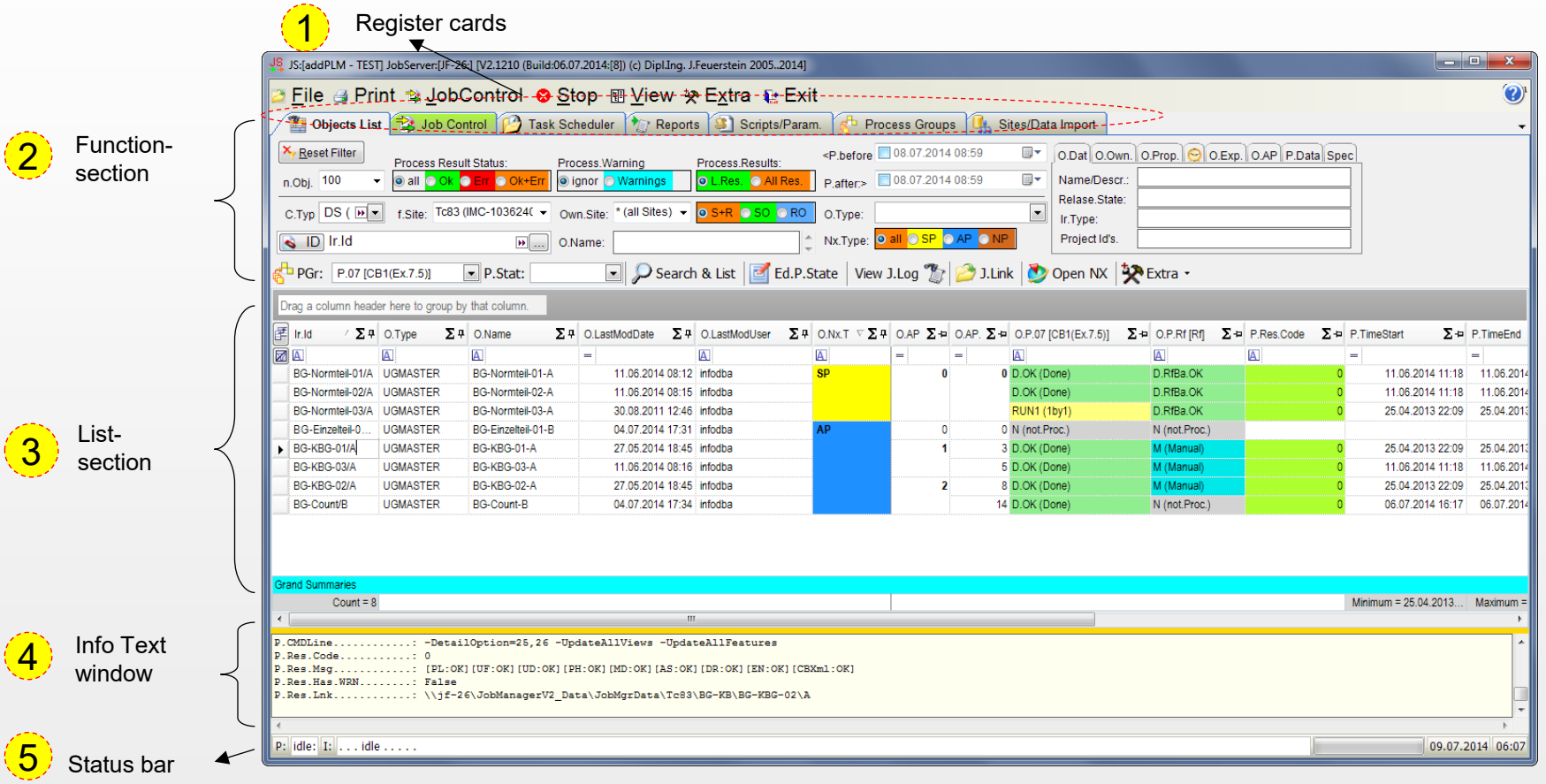
- **Objects List** → View of Object Data and JobResults
- **Job Control** → Organizes JobClients and Job Processes
- **Task Scheduler** → Controls time related processes
- **Reports** → Generates Reporting of Object and Job Statistic and Results
- **Scripts/Param.** → Defines and organizes Job Scripts – Parameters
- **Process Groups** → Defines the Job Groups
- **Sites/Data Import** → Registers Customer Sites and Imports Site Object Data



Objects List

Basics

The GUI of JobServer is divided by the register cards (1) and each of them has different main tasks. Each of the register cards has the following sections: Functions section (2) List section (3) Info window section (4) and Status bar (5). The Concept of PLMJobManager is to have in all register cards a most likely similar way of handling the Software.



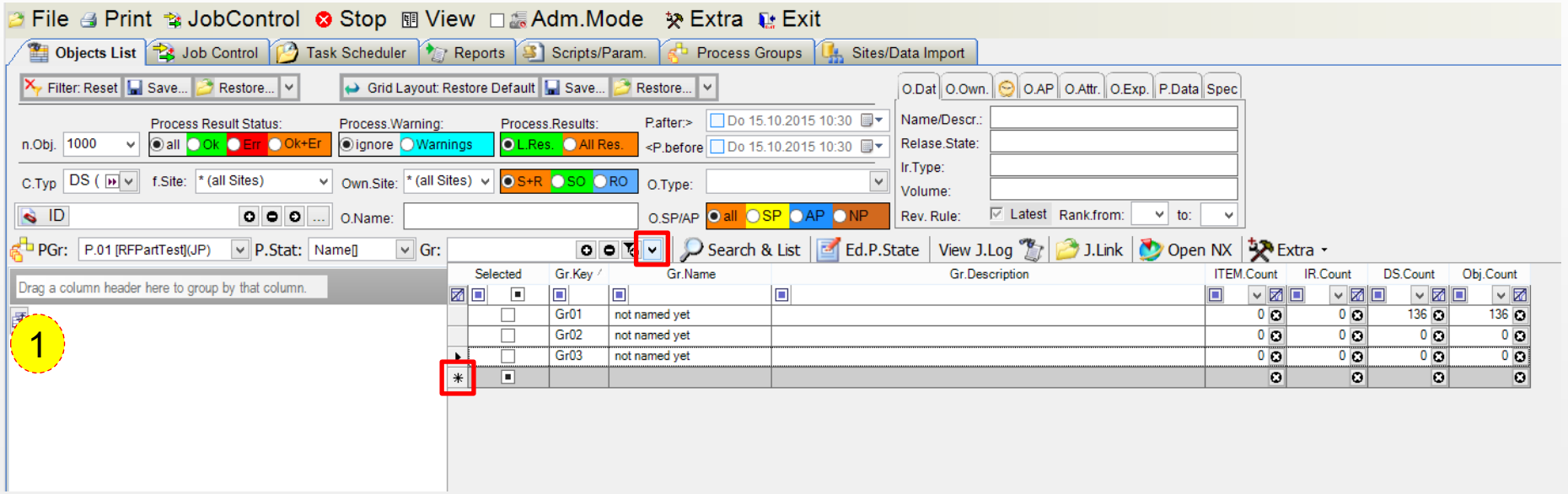
Objects List: using Object Groups

- Create a group in JobManager

Click on the arrow sign at the group combo box

Click on the star sign to create a new group

Specify Gr.Name and Gr.Description



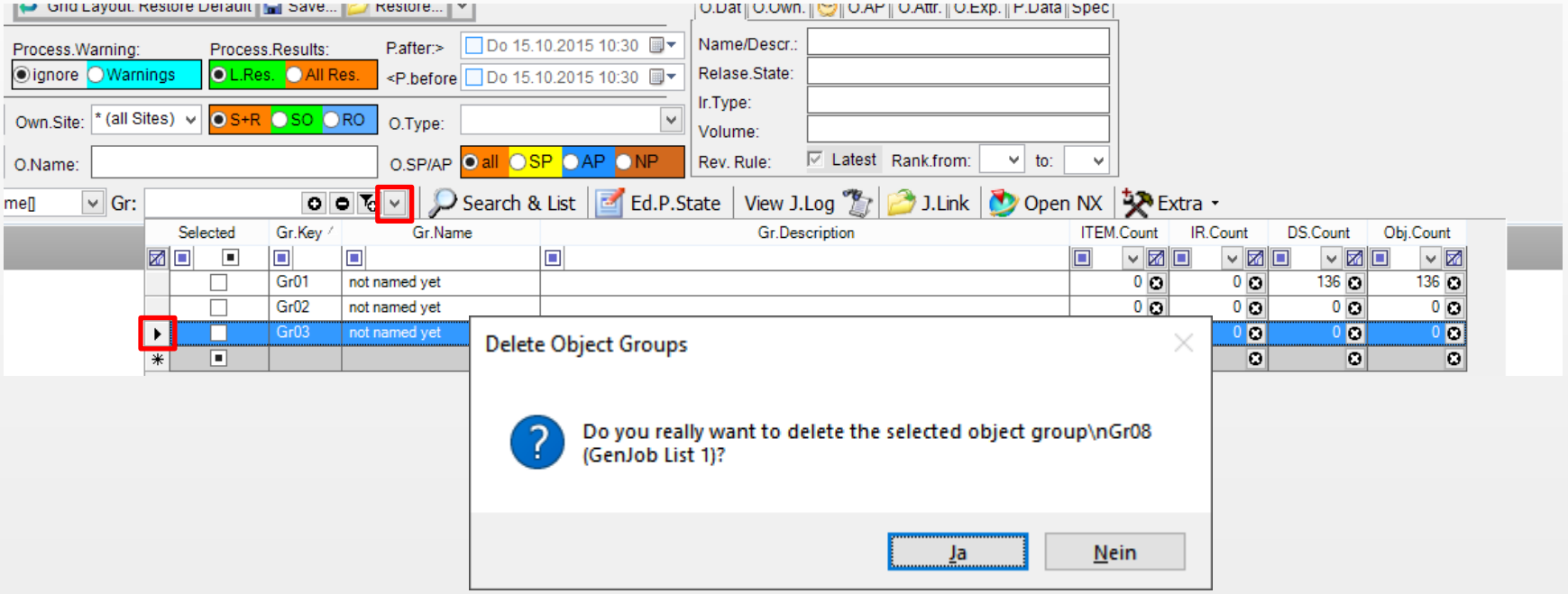
Objects List: using Object Groups

- Delete a group in JobManager

Click on the arrow sign at the group combo box

Select the whole row of the group you want to delete

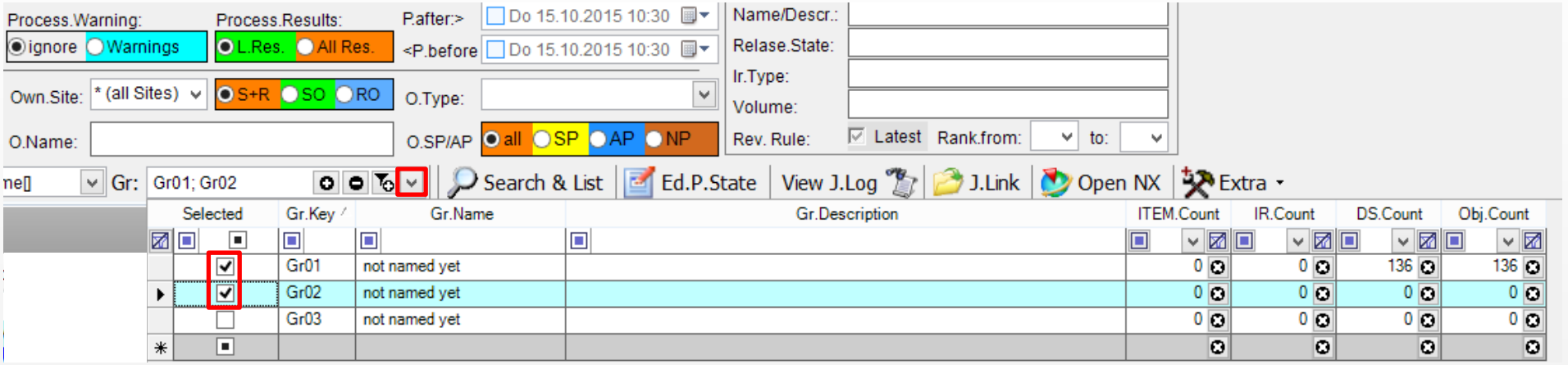
Hit the delete key on your keyboard → Delete Object Groups Dialog is shown



Objects List: Using Object Groups

- Add objects to a group in JobManager 1/3

Click on the arrow sign at the group combo box
Select the groups you want to add objects to (this can also be multiple groups)



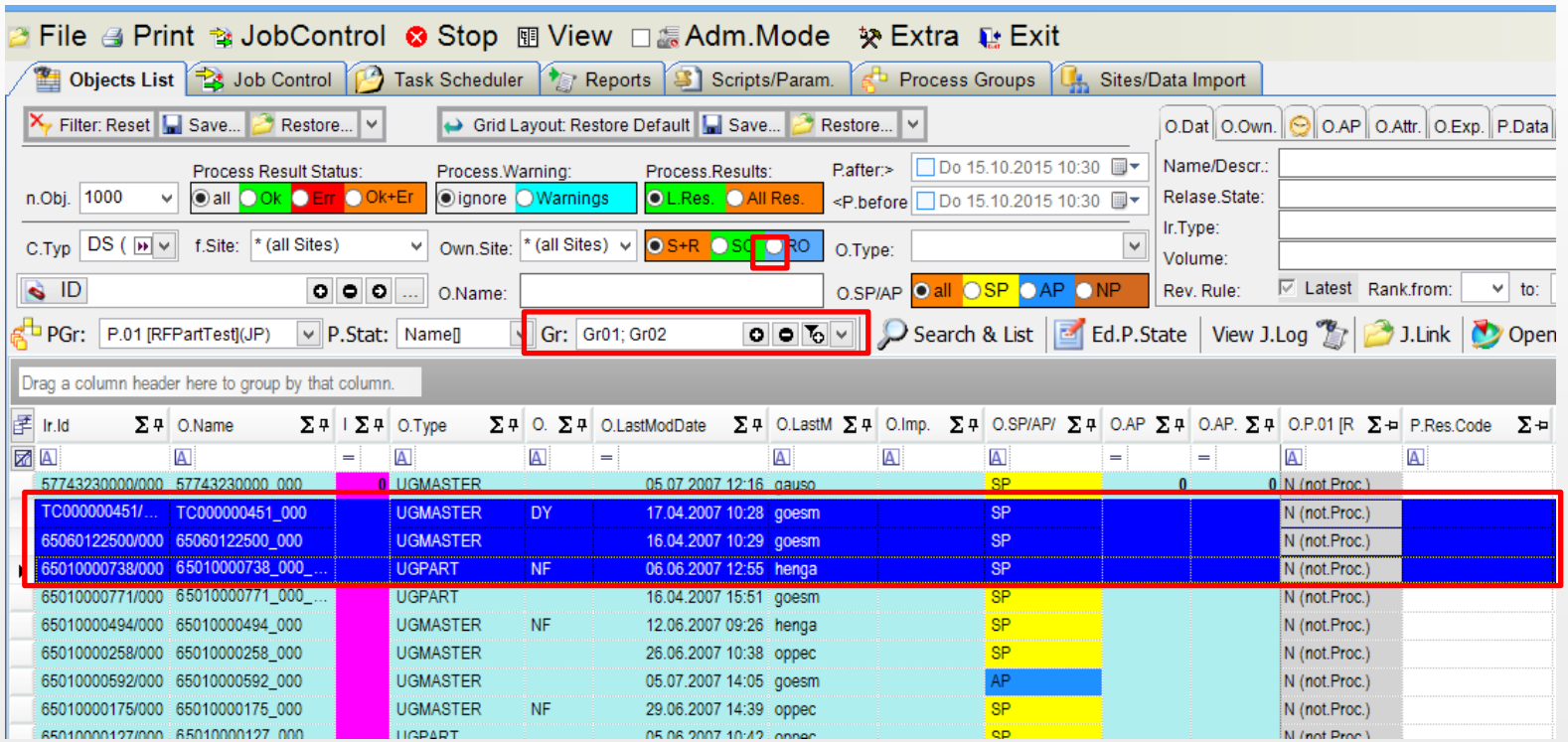
Objects List: Using Object Groups

- Add objects to a group in JobManager 2/3

There are 2 ways to add objects to the group(s)

The first is to select objects in the data grid and then to hit the + button in the Gr: field
(you can select the whole row or even single cells inside the grid)

Only the selected objects will be added to the group (only when they were not in the group before)



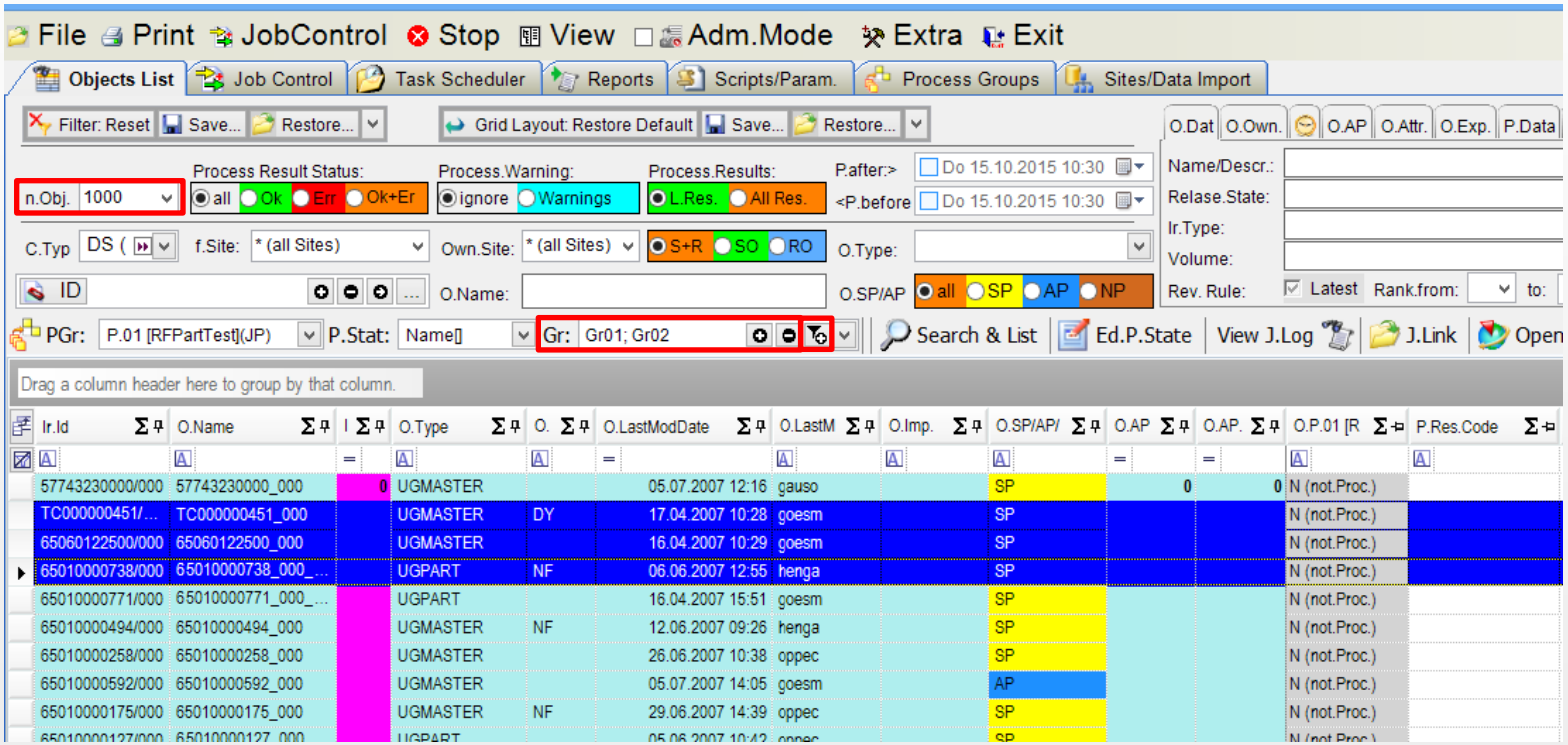
In this case the selected 3 objects will be added to the groups Gr01 and Gr02

Objects List: Using Object Groups

- Add objects to a group in JobManager 3/3

The second way to add objects to the group(s):

Click the “filter+” sign in the Gr: field
All objects that will be find by the current specified filter will be added to the group(s)s
(Without displaying the objects in the grid)

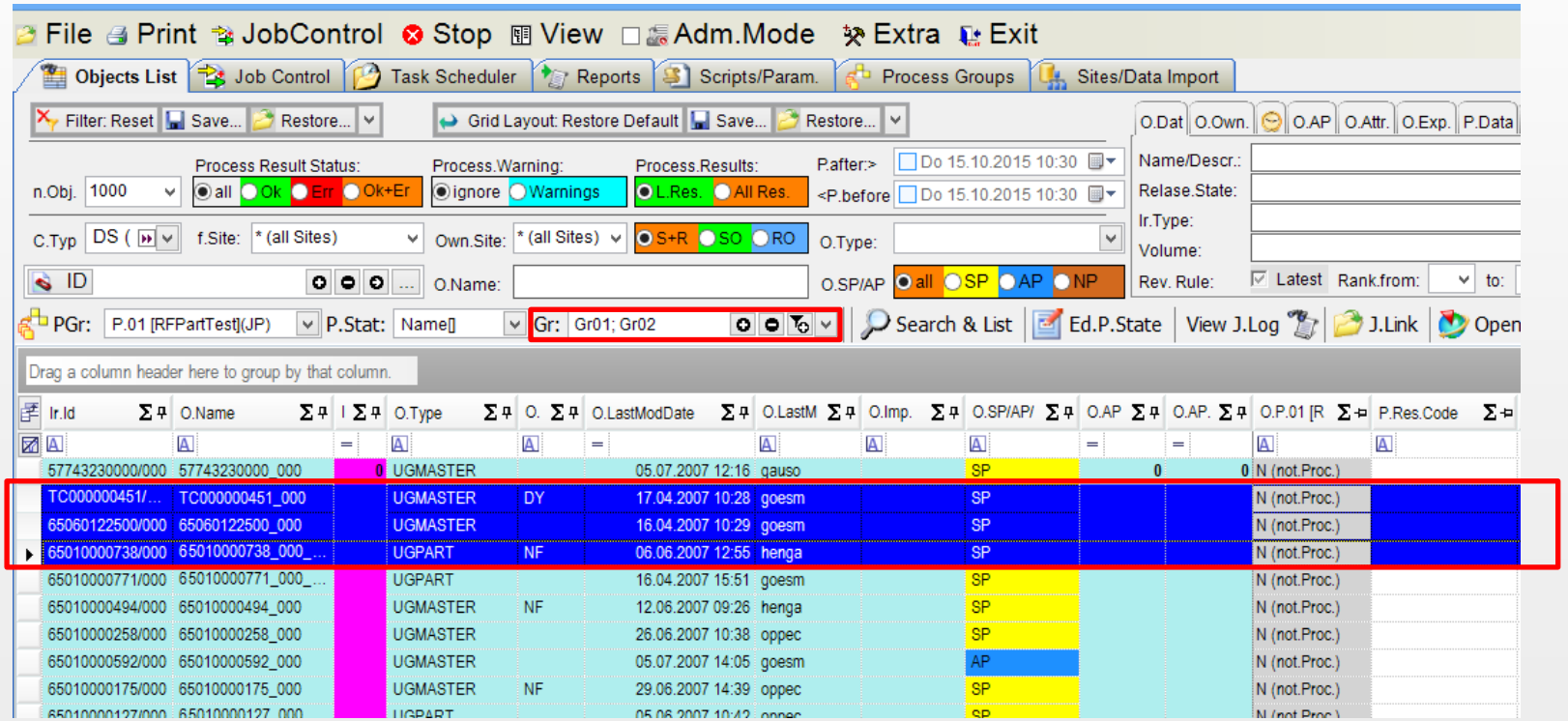


In this case 1000 random objects will be added to the groups Gr01 and Gr02
(because the filter is specified only to get 1000 random objects)

Objects List: Using Object Groups

- Remove objects from a group 1/2

There are 2 ways to delete objects from (a) group(s)
The first is:
Select the group(s) you want to delete the objects from
Select objects in the data grid and then hit the - button in the Gr: field (you can select the hole row or even single cells inside the grid)
Only the selected objects will be removed from the group(s) (only when they were in the group before)

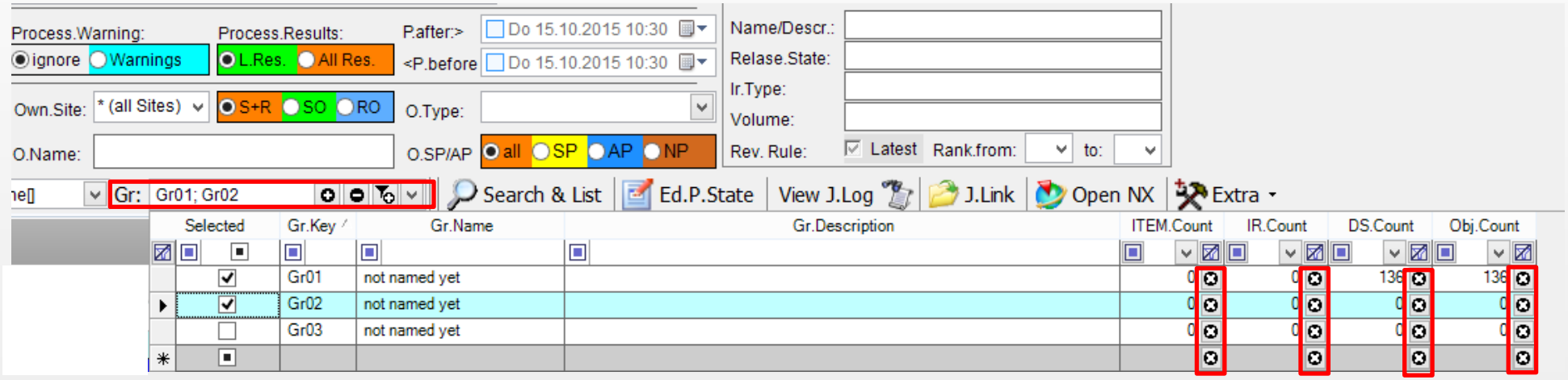


In this example the selected 3 objects will be removed from the groups Gr01 and Gr02

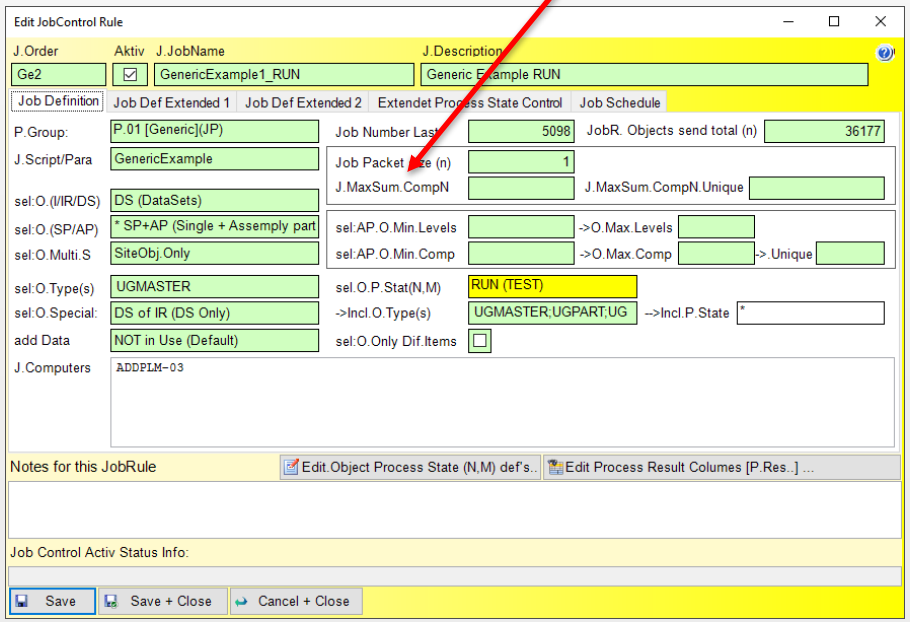
Objects List: Using Object Groups

- Remove objects from a group 2/2

The second is:
Click on the arrow sign at the group combo box
Click one of the x buttons in the row of the group you want to delete the objects from
Clicking the x button on the “ITEM.Count” column will delete all item objects from the group
Clicking the x button on the “IR.Count” column will delete all item rev objects from the group
Clicking the x button on the “DS.Count” column will delete all dataset objects from the group
Clicking the x button on the “Obj.Count” column will delete all objects from the group



Edit JobControl Rule



Job Max Sum N Comp

This Number defines how many Components (O.Ap.CompN) may in sum contained in the JobPacket.
Therefor it sum up the number of the Assembly components (O.Ap.CompN) and stops
adding components to the JobPacket if the given limit is reached.

Special Rule:
Every Job contains at least 1 Object

UseCase:
This Rule can be used to limit the mound of Assemblies contained in the Job.

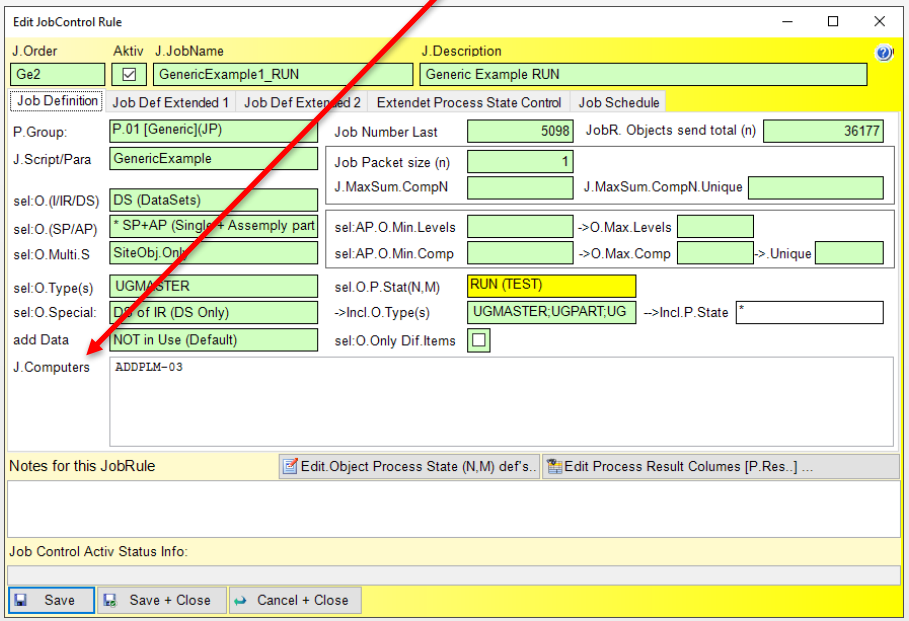
Example1: Defined Limit = 100

IR	O.AP.CompN	JobSumComp	add to Job
4711/001	10	+ 0 = 10	YES
6523/001	30	+ 10 = 40	YES
7891/003	50	+ 40 = 90	YES
8913/001	60	+ 90 = 150	NO --> Limit 100 reached
5712/001	70	+150 = 220	NO --> Limit 100 reached

Example2: Defined Limit = 45

IR	O.AP.CompN	JobSumComp	add to Job
4711/001	10	+ 0 = 10	YES
6523/001	30	+ 10 = 40	YES
7891/003	50	+ 40 = 90	NO --> Limit 45 reached

Edit JobControl Rule



JobRule JobClient settings

Defines list of JobClients that get acces to this JobRule
and defines optional how many Job's may run on a JobClient

```
Syntax......: Computer1[:n][:NoFurtherJob];ComputerXY*[:n][:NoFurtherJob]

#Example 1..: *
#Description.: Rule is used for all Computers

#Example 2..: Computer1;Computer2;CompNameXY*
#Description.: |-> Rule is used for Computer
#Description.:     with Name Computer1;Computer2; and like CompNameXY*

#Example 3..: *:1
#Description.: |-> All Computer runs 1 Job of JobRule

#Example 4..: Computer1:5;Computer2:2
#Description.:  |          |-> Computer2 runs max 2 Jobs of JobRule
#Description.:  |-> Computer1 runs max 5 Jobs of JobRule

#Example 5..: Computer1:1:NoFurther;JobComputer2:5:NoFurtherJob
#Description.:  |-> Behavior like 'Example 2'
#Description.:      but if computer runs n JobMax then no
#Description.:      further job is searched in other JobRules
```

Edit JobControl Rule

- Steuerung des Prozess Ergebnisses (‘ProcStateValueRule‘)

Definieren der : ‘ProcStateValueRule‘

Über die ProcStateValueRule können Regeln festgelegt werden wie der P.Status Wert in Abhängigkeit vom Job Ergebnis sich festlegt.
Ausgewertet wird dabei die Daten des JobErgebnisses: ResultCode ResultMessage ResultWarning

ResultMessage: (Key:O.Type) Optional
werte den Object Typen (UGMASTER,UGPART,UGALTREP..) aus es ermöglicht das

ResultCode: (Key:J.ResCode) Erforderlich
es kann definiert werden: RC=0 RC>0 RC<>0 RC<0

ResultMessage: (Key:J.ResMsg) Erforderlich
Die der Wert der JobMessage wird via Win Like Methode verglichen Beispiel: RM=* oder RM=*Corba*

ResultWarning: (Key:J.ResWarn) Erfoderlich
Der Wert von JobWarning (Boolean) 0=false 1=true verglichen ResWarn=* (true da Ignoriert) RW=1 (true has Waring) oder RW=0 (false no Warning)

Result Process Status: (Key:P.State)
Definiert den Wert den das Objekt erhält wenn eine der Regeln zutrifft.
Die Auswertung der ProcStateValueRule wird beendet sowie eine der Regeln zutrifft. Der das Ergebnis P.Stat=… wird dann in dann als P.Status übergeben.
Des weiteren gilt das wenn keine der Regeln zutreffen P.Status = D (Done) gesetzt wird

Beispiel Syntax der Definition:

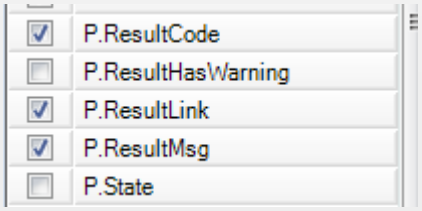
Jede Definition steht in [] Klammern und kann aus mehreren Regeln bestehen
Default Values: „..\01-BinServer\Resources\ApplicationStandards\ProcStatValueRule_Template.TXT“

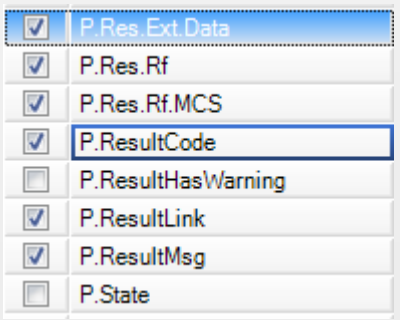
```
[RuleID=Default] [P.Stat=D]
[RuleID=OK]      [O.Type=UGMASTER]  [J.ResCode=0] [J.ResMsg=*] [J.ResWarn=0] [P.Stat=D.RfBa.OK]
[RuleID=OK]      [O.Type<>UGMASTER] [J.ResCode=0] [J.ResMsg=*] [J.ResWarn=0] [P.Stat=D.OK]
[RuleID=Err]     [O.Type=*]         [J.ResCode>0] [J.ResMsg=*] [J.ResWarn=*] [P.Stat=D.RfBa.ERR]
[RuleID=Warn]    [O.Type=*]         [J.ResCode=*] [J.ResMsg=*] [J.ResWarn=1] [P.Stat=D.RfBa.W]
```

Definition Process Result Erstellung (New 02.11.2013)

- Steuerung des Prozess Ergebnisses

```
Definition der Ergebniss bestimmung  Generische festlegung

Daten P.Res.MSG
[KEY:STA Description]  ResultCode
 |   |   |              |-> Interpretation ResultCode="0" OK
 |   |   |              |-> Interpretation ResultCode="0" WRN
 |   |   |              |-> Interpretation ResultCode<> "0" ERR
 |   |   |-> Result Short Description
 |   |-> Result Status OK ERR WRN
 |-> Result Key

Daten P.ResResult.MSG
Daten P.Res.Ext.Data
[MD:ERR MomOfIn]

Daten in Spalten: P.Res.MSG.Ext.Data
[MD:ERR MomOfIn:e-1]
[MD:ERR MomOfIn:e-2]
                |-> ResultValue  e-1 e-2 #=#
```





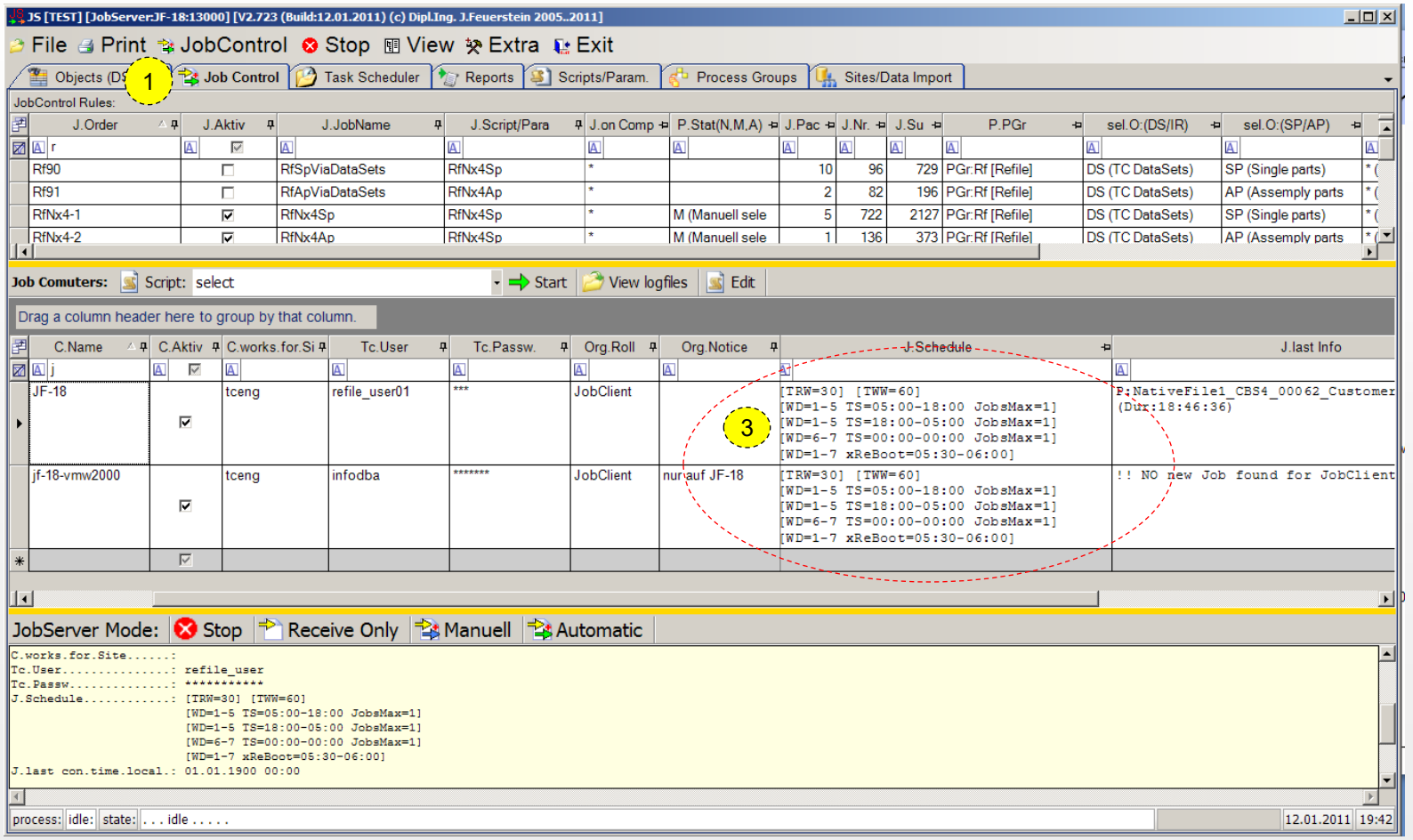
Neue Benennung:

P.Res.Msg
P.Res.Msg.Ext.Data
P.Res.Has.WRN
P.Res.Lnk
P.Res.State

Job Control -> Job Computers -> ‚J.Schedule‘ Zeiten via JobServer steuern

- Steuerung der JobClient Schedule Zeiten

Der JobServer steuert die Zeiten in dem ein JobClient aktive ist über JobControl (1) → Job Computers **(2)** J.Schedule **(3)**



Job Control -> Job Computers -> ‚J.Schedule‘ Zeiten via JobServer steuern

Steuerung der JobClient via ‘ **JobClientScheduleControl‘**

Wochentag Steuerung (Key:WD WeekDay):
[WD=*] alle Tage der Woche
[WD=1-7] alle Tage der Woche
[WD=1-3] alle Tage der Woche Montag – Mittwoch
[WD=1-3-Montag-Mittwoch] die Tage von Montag bis Mittwoch also inc Dienstag
[WD=6-7-Sa-So] die Tage von Samstag und Sonntag
[WD=3-Mi-Fron] nur der Mittwoch

Zeitpunkt Key:TP TimePoint:
[WD=1-7 TP=21:13] wird dann als OK zurückgegeben wenn die aktuelle Zeit in Std und Min übereinstimmt
TWR= Dauer in Sec die der JobClient warten soll wenn er innerhalb der Zeit liegt bevor er sich wieder mit dem Server verbindet Beispiel: [TWR=30]TWW= Dauer in Sec die der JobClient warten soll wenn er außerhalb der Zeit liegt bevor er sich wieder mit dem Server verbindet Beispiel: [TWW=60]

Zeitspanne: Key:TS TimeSpan =TS.Begin-TS.End
[WD=* TS=00:00-00:00] [WD=1-Mo TS=20:30-05:00]
[WD=1-3 TS=10:30-03:00] [WD=4-7 TS=08:30-03:00]
[WD=1-5-Mo-Fr TS=19:15-05:30] [WD=3-Mi-Fron TS=19:15-23:59] [WD=4-Do-Fron TS=00:00-17:30] [WD=6-7-Sa-So TS=00:00-00:00]

Define a Redo interval: Key:Redo=n sec‘s // #New 14.08.2014
[WD=* TS=00:00-00:00 ReDo=60] Redos the Job 24h every 60 seconds
[WD=1-5 TS=08:00-17:00 ReDo=120] Redos the Job in weekdays Monday .. Friday in times 08:00 until 17:00 every 120 seconds

Anzahl Jobs (Key:JobsMax short Key:JMax):
definiert wie viele Jobs auf dem Client im angegebenen Zeitbereich gleichzeitig abgearbeitet werden dürfen.
Hinweis: Wird mit JobsMax=0 angegeben so wird KEIN Job im angegebene Zeitbereich verarbeitet **Beispiel.:** [WD=1-5 TS=07:00-17:30 **JobsMax=0**]
Beispiel: [WD=6-7 TS=00:00-00:00 **JobsMax=2**] definiert das am Samstag und Sonntag zu allen Zeiten maximal 2 Jobs auf dem Client laufen

Job Resoure Steuerung (Key:JobsResPointsMax short Key: JrpMax):
definiert wie viele Resoure Punkte ein Client hat. Diese Zahl wird verglichen mit der Anzahl der Resourepunkte die sich aus der Summe der Resourepunkte der auf dem Clienten laufenden Jobs ergeben (**ResourcePointsOfCurExecutingJobs**). Wenn ResourcePointsOfCurExecutingJobs+JobResourcePoints>JobsResPointsMax so erhält dieser Client keine Jobs
Beispiel: [WD=6-7 TS=00:00-00:00 JobsMax=2 JobsResPointsMax=2]

Der JobClient Computer automatisch neu starten (Key:Reboot:)
Beispiel: [WD=1-7 ReBoot=05:30-06:00] definiert das sich der Rechner an den allen Wochentage in der Zeit zwischen 05:30 und 06:00 bootet wenn in dieser Zeitspanne keine Job mehr bearbeitet wird.

Job Control -> Job Computers -> ‚J.Schedule‘ Zeiten via JobServer steuern

- Steuerung der JobClient Schedule Zeiten

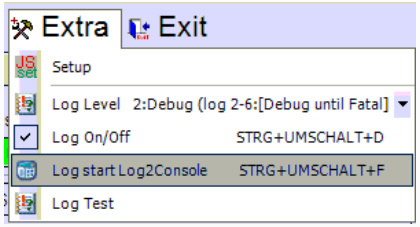
Example: Standard settings:

```
[TRW=30] [TWW=60]
[WD=1-5 TS=05:00-18:00 JobsMax=1 JobsResPointsMax=-1]
[WD=1-5 TS=18:00-05:00 JobsMax=1 JobsResPointsMax=-1]
[WD=6-7 TS=00:00-00:00 JobsMax=1 JobsResPointsMax=-1]
[WD=1-7 xReBoot=05:30-06:00] Hinweis wird xReBoot=… angegeben so wird ReBoot ignoriert
```

JobServer: Debuging JobManager Software

- Working with Debug window „Log2Console“ This Tool can be used for debugging the PLMJobManager software programs

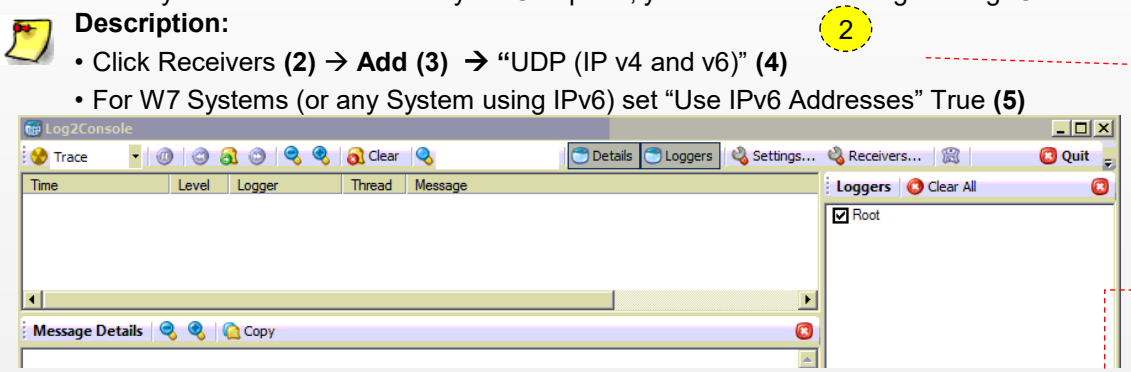
Start the Software via Extra → Log start Log2Console (STRG+UMSCHALT+F) **(1)**

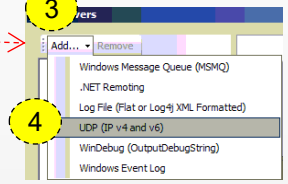


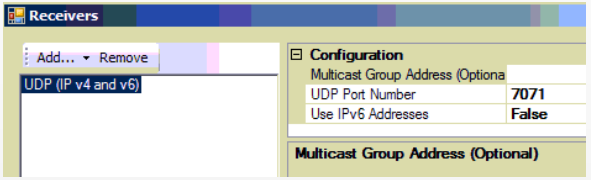
First time you use the Software on your Computer, you have to set settings in Log2Console Application:

Description:

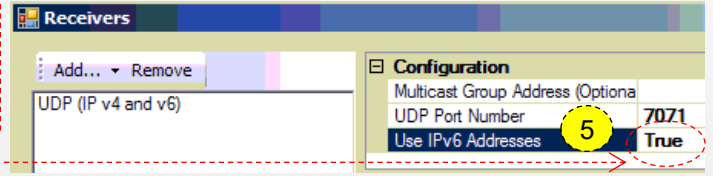
- Click Receivers **(2)** → **Add (3)** → “UDP (IP v4 and v6)” **(4)**
- For W7 Systems (or any System using IPv6) set “Use IPv6 Addresses” True **(5)**



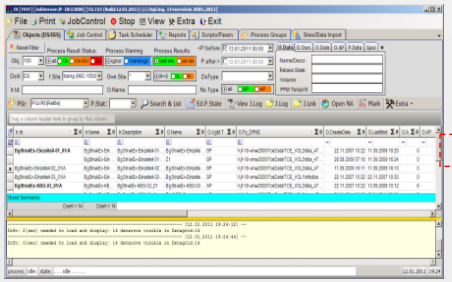


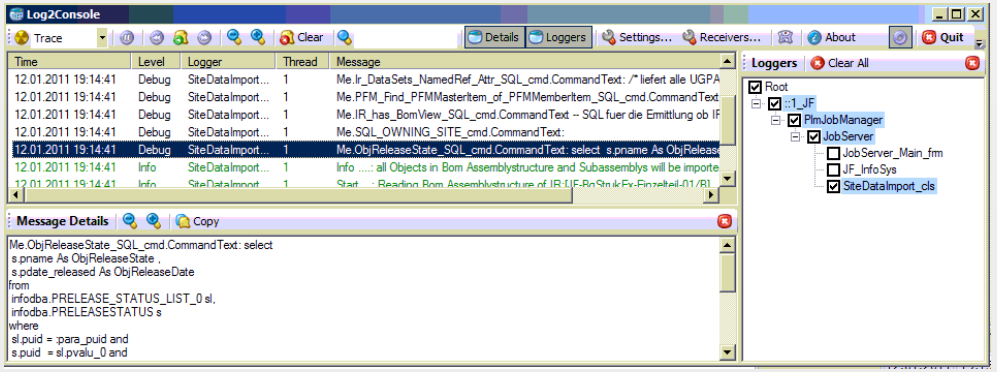


Set on Win7 Use IPv6 Adresses to **True**



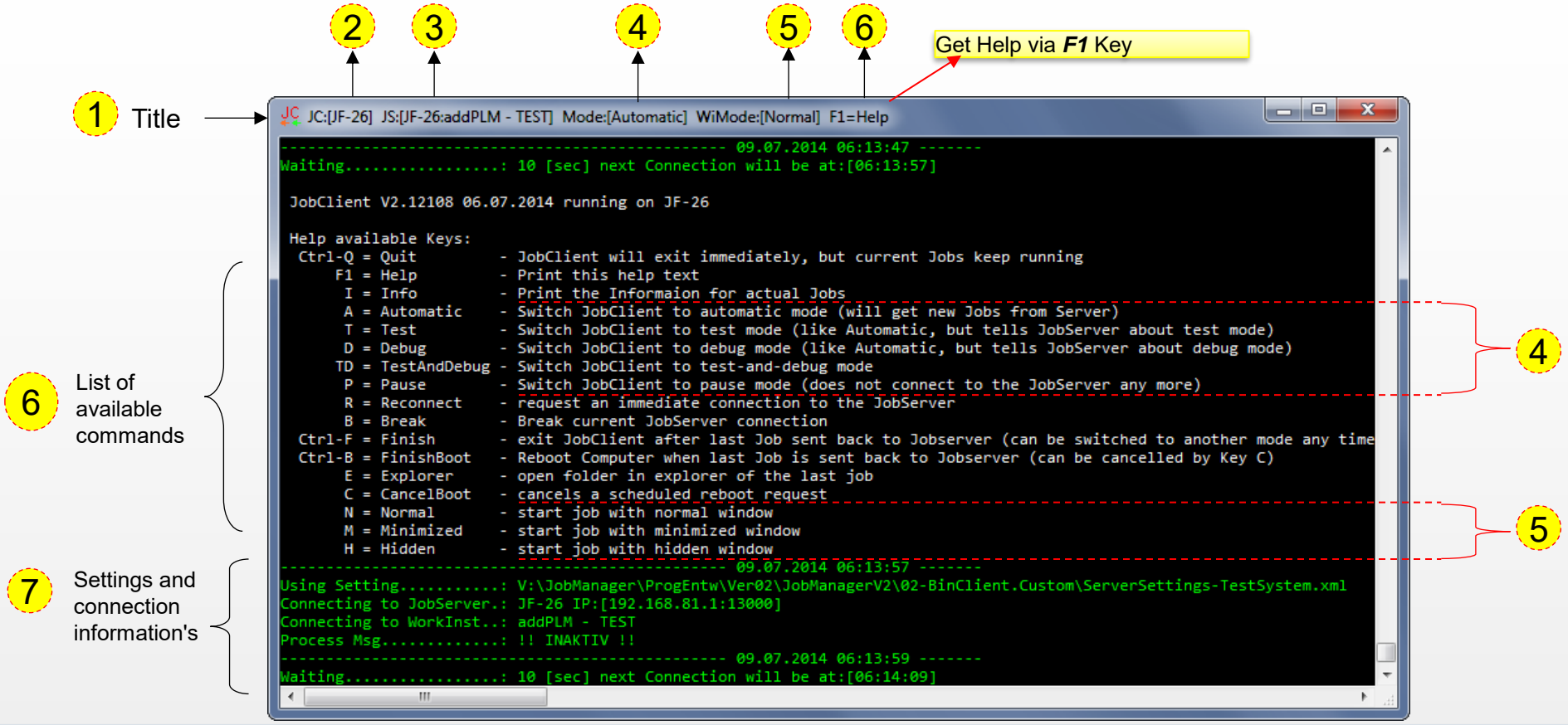
After setting up the software JobServer JobClient or JobPlaner is sending Log information's to Log2Console Software





JobClinet

- The GUI of JC has a style of a command line tool.

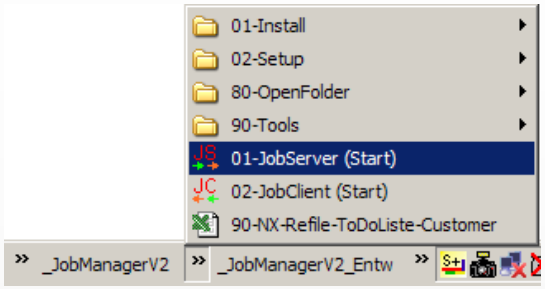


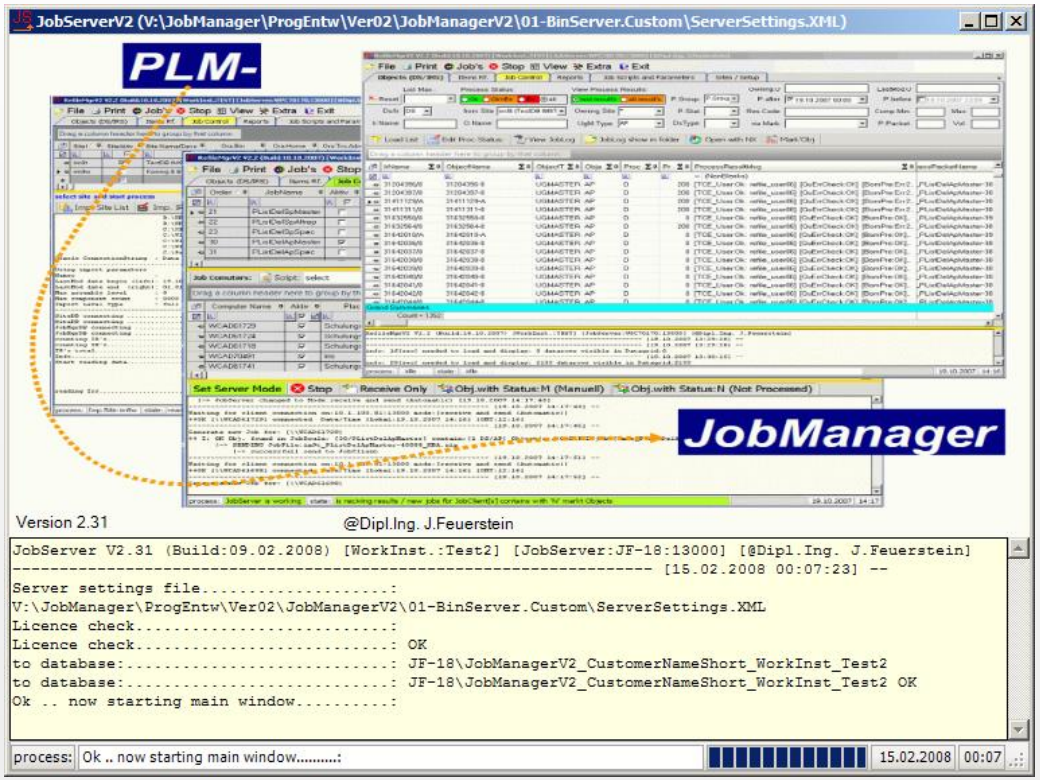
Titel **(1)** of JC shows Name of JC Computer **(2)**. Name of JobServer **(3)** the JC is communicating with. **(4)** Shows the current mode of JC Modes can be Automatic Test Debug TestAndDebug Pause. **(5)** Shows current WinMode of new created Job Processes. Via Key ***F1*** you will see all available commands **(6)**. Used settings and connection information's are available on bottom of command line window **(7)**.

JC communicates with the JobServer via tcp socket connections.

Setup / Config JobServer and Processes

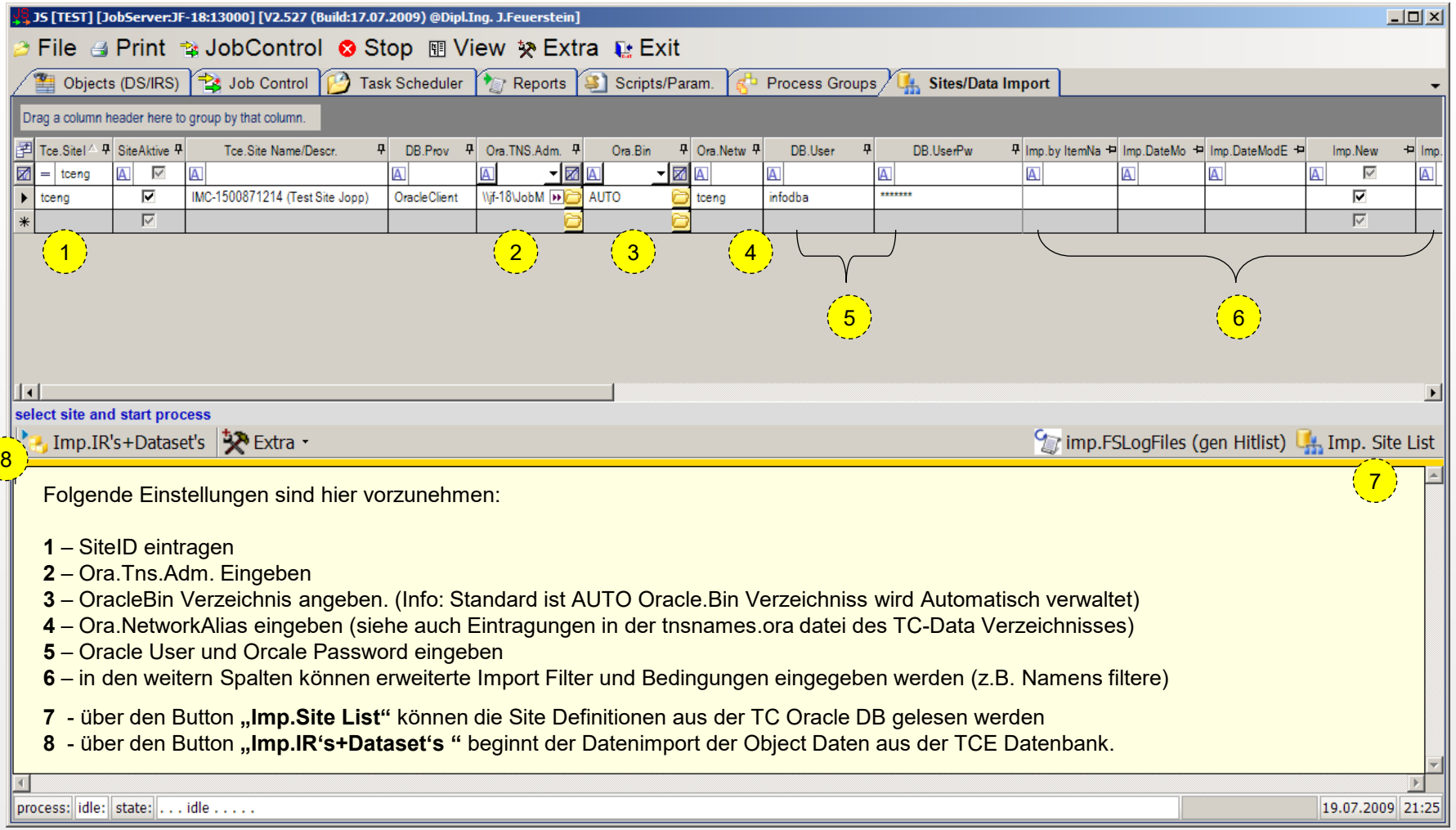
- Starten des JobServers





Setup / Config JobServer and Processes

- Sites/Data Import aus Oracle TCE Datenbank



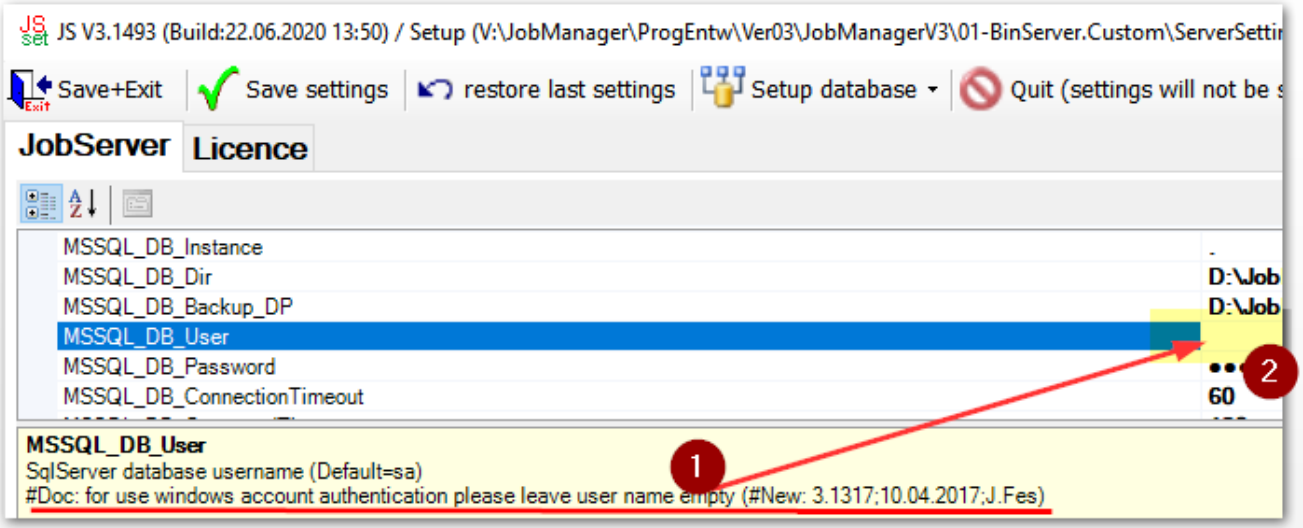
Example Connection String for MS SQL Server DB:

Data Source=MyServer\MyTcMsqlDbServerName;Initial Catalog=MyTcDb;User ID=infodba;Password=infodba;Connection Timeout=60;Persist Security Info=True;MultipleActiveResultSets=True;

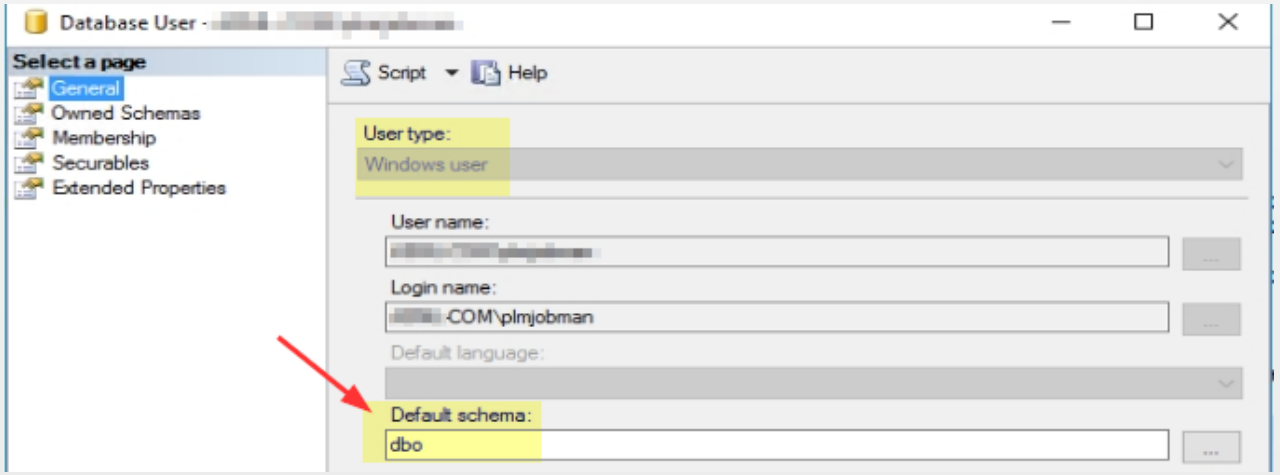
Setup / Config JobServer and Processes

- JobManager using Windows Audentification

When JobManger is using windows login method then please setup:

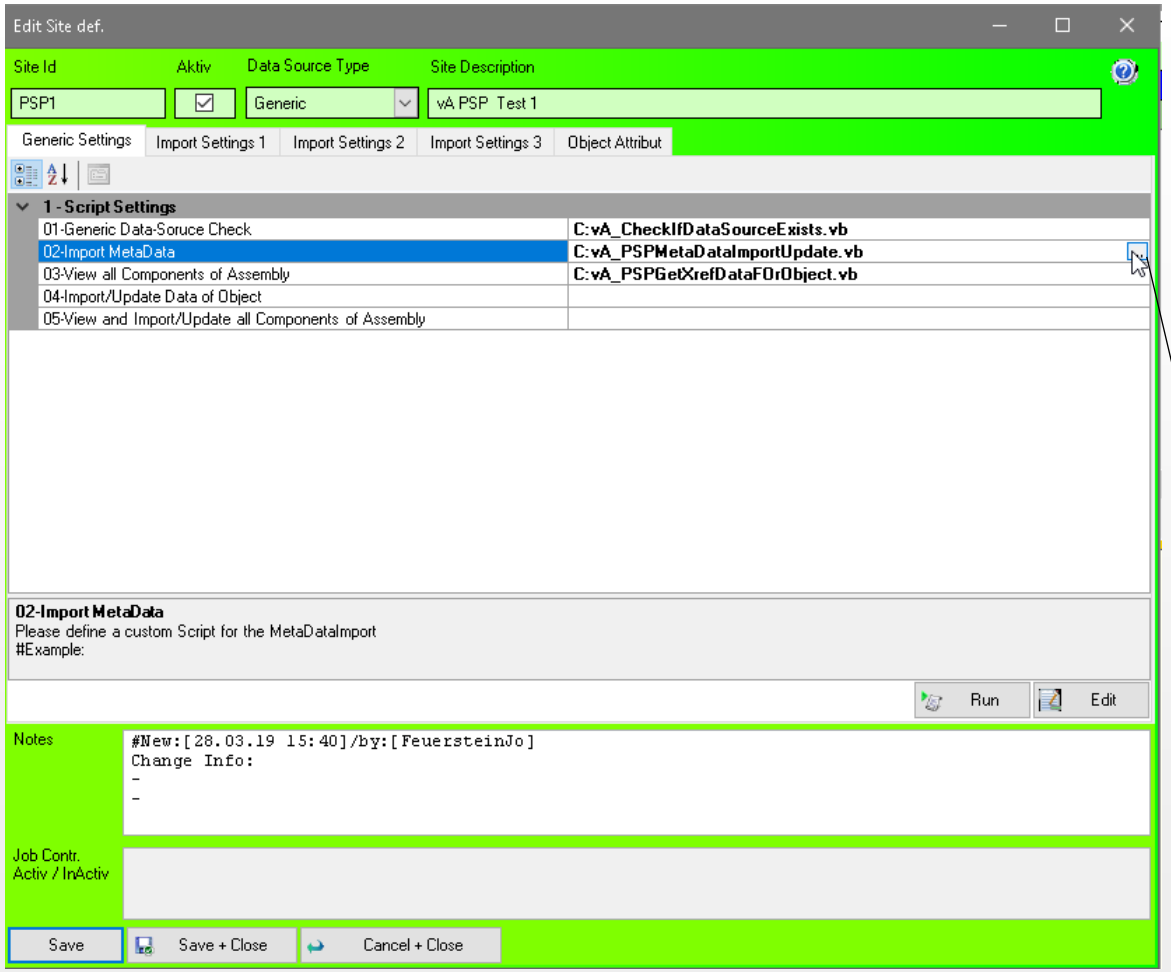


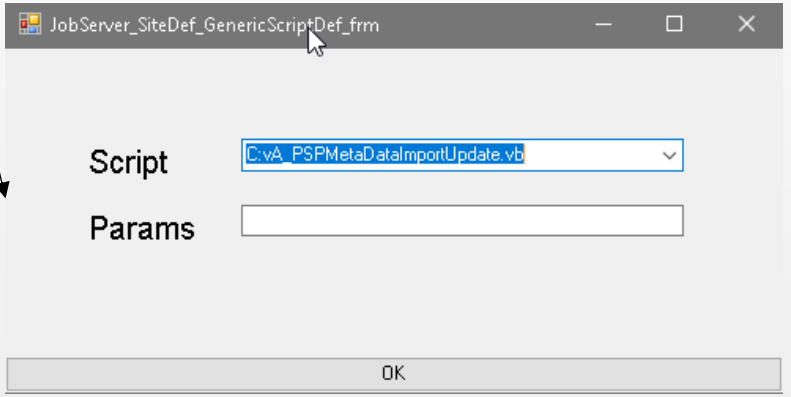
The default schema need to be assigned to dbo



Setup / Config JobServer and Processes

- Sites/Data Import aus generischen Datenbanken





Example Connection String for MS SQL Server DB:

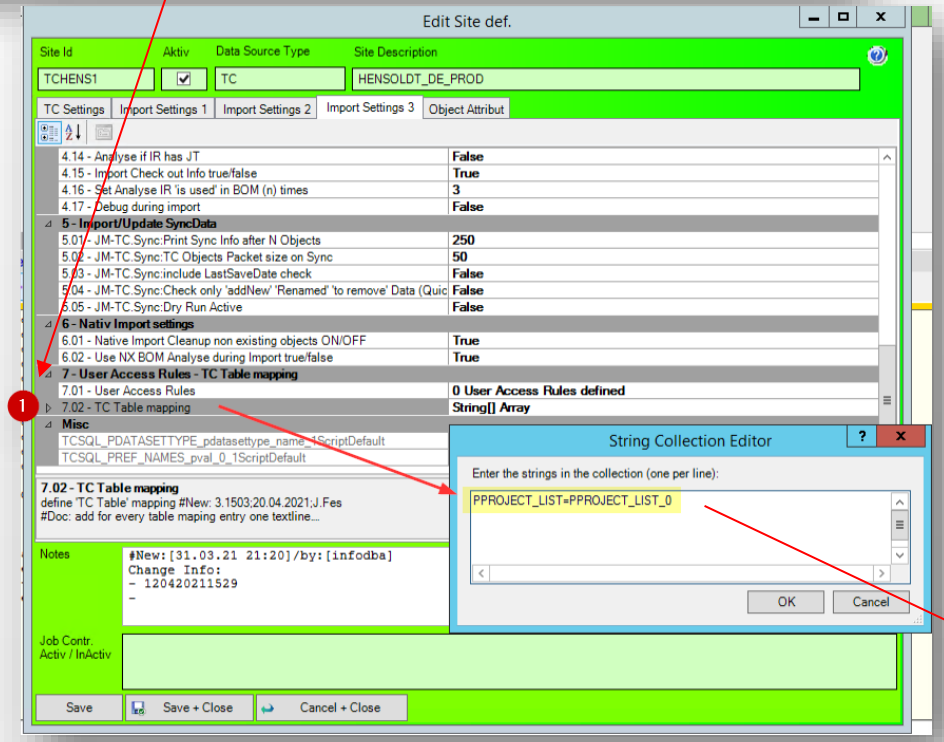
Data Source=MyServer\MyTcMsqlDbServerName;Initial Catalog=MyTcDb;User ID=infodba;Password=infodba;Connection Timeout=60;Persist Security Info=True;MultipleActiveResultSets=True;

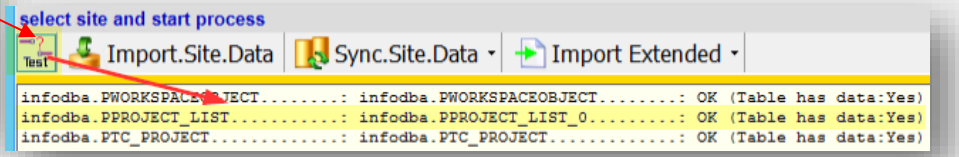
Edit Site def. → 7.02 TC Table Mappings

addPLM GmbH
Web: www.addPLM.com

- It is now possible to define TC Table mappings.

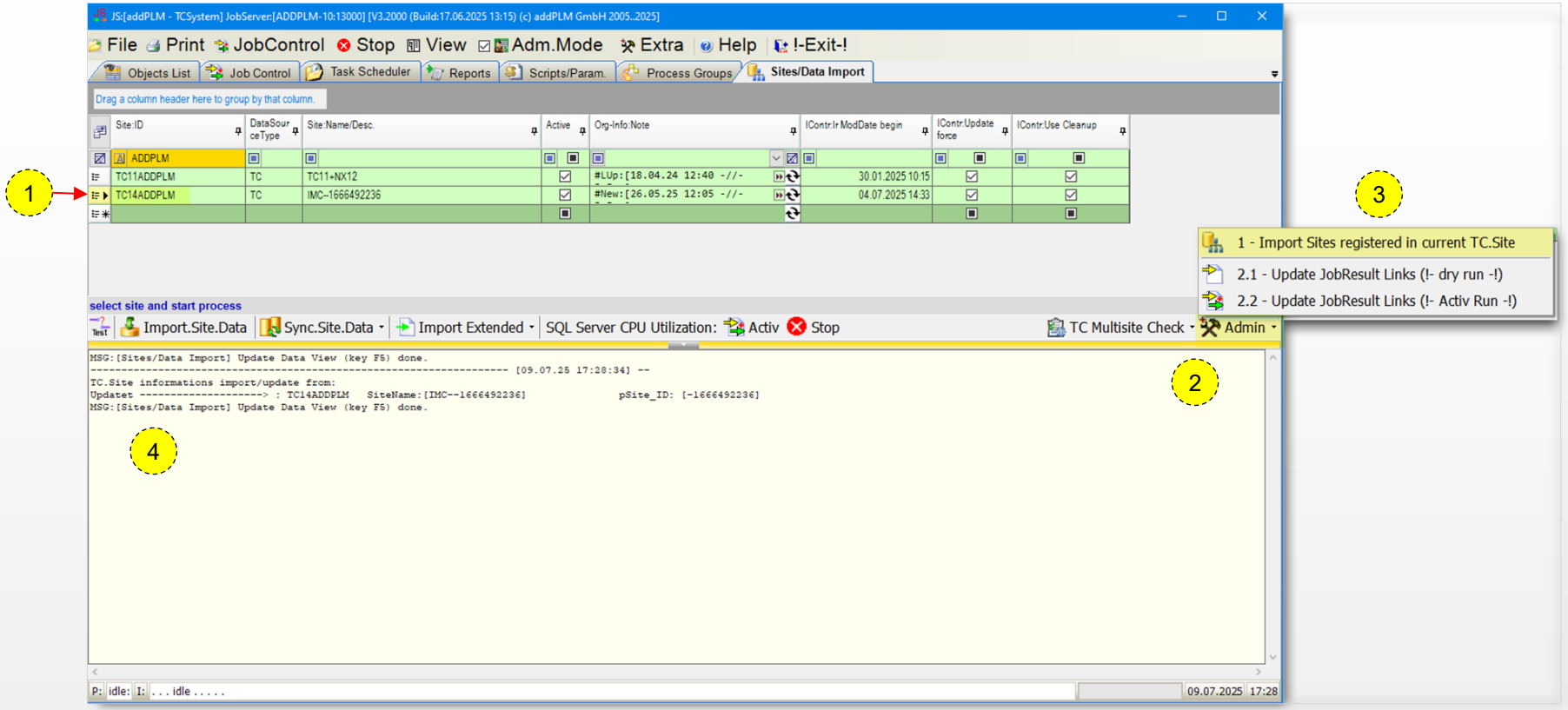
```
=== 21.04.2021 ================================================================================
- Version.........: 3.1503 // J.Fes
- Categorie.......: Import Sites/Data Import Settings
- Additional Info: Ver.Info Check.Settings
- Type............: New:
- Description....: JobServer TC.Site definitions enhanced
                   It is now posible to define TC Table mappings.

                   Please see settings:
                   Sites/ Data Import -> SiteDef -> 7.02 - TC Table mapping

                   Settings can be used to map to correct project table via define setting
                   PPROJECT_LIST=PPROJECT_LIST_0

                   Enhancement implemented by code extensions
```





Setup / Config JobServer and Processes

- Sites/Data Import

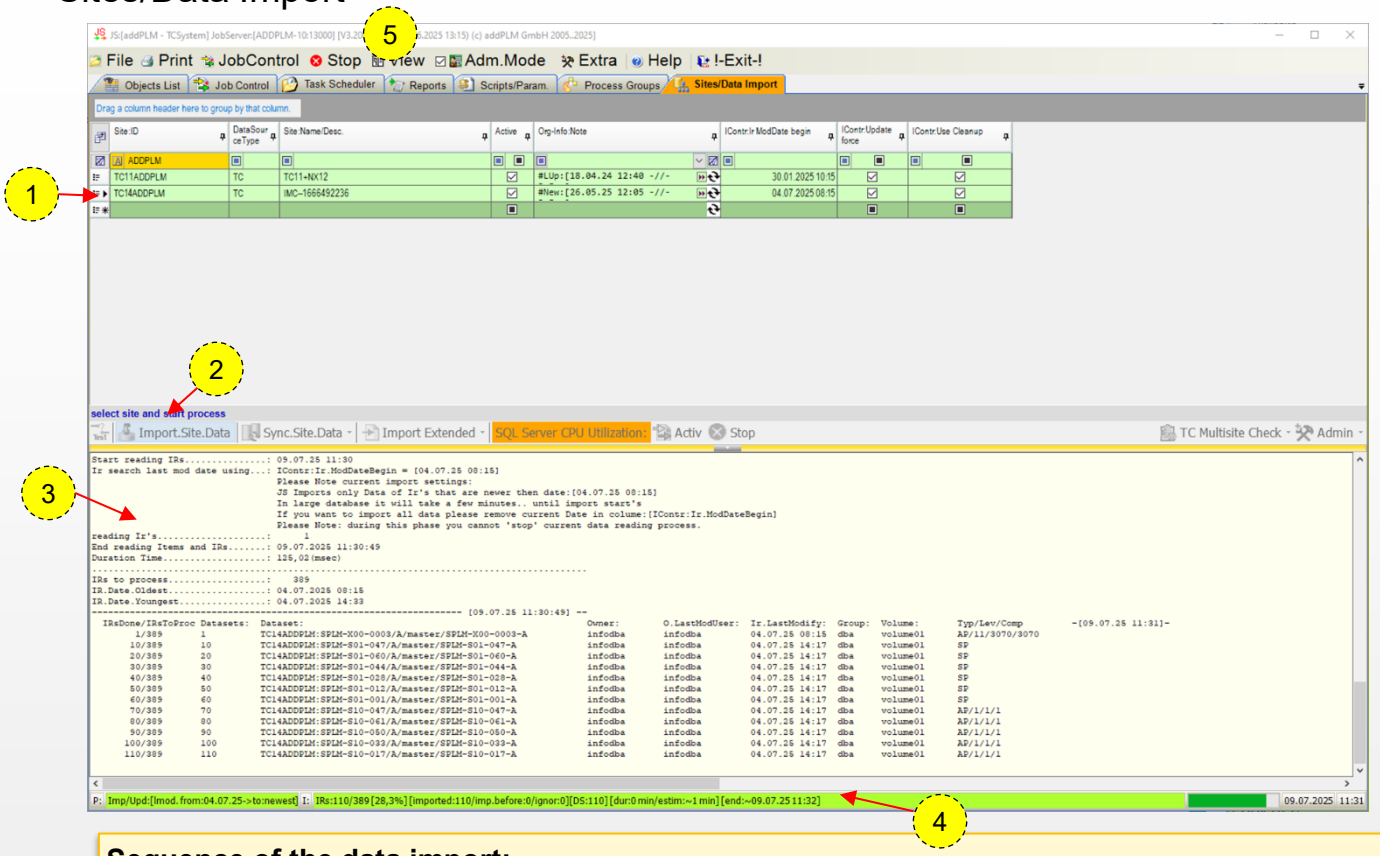


Procedure:
1 – Select the site def by selecting the row or by clicking in one of the cells
2 – Select Admin
3 – Click on button “Import Sites registered in current TC.Site" -> Starts importing the site definitions
4 – Output of imported Side definitions

Note:
If the TC database is a multisite database, all site definitions are recorded and listed.

Setup / Config JobServer and Processes

- Sites/Data Import



Sequence of the data import:

1 - Select the site def by selecting the row or by clicking in one of the cells

2 - Click on the 'Import.Site.Data' button → Start the data import

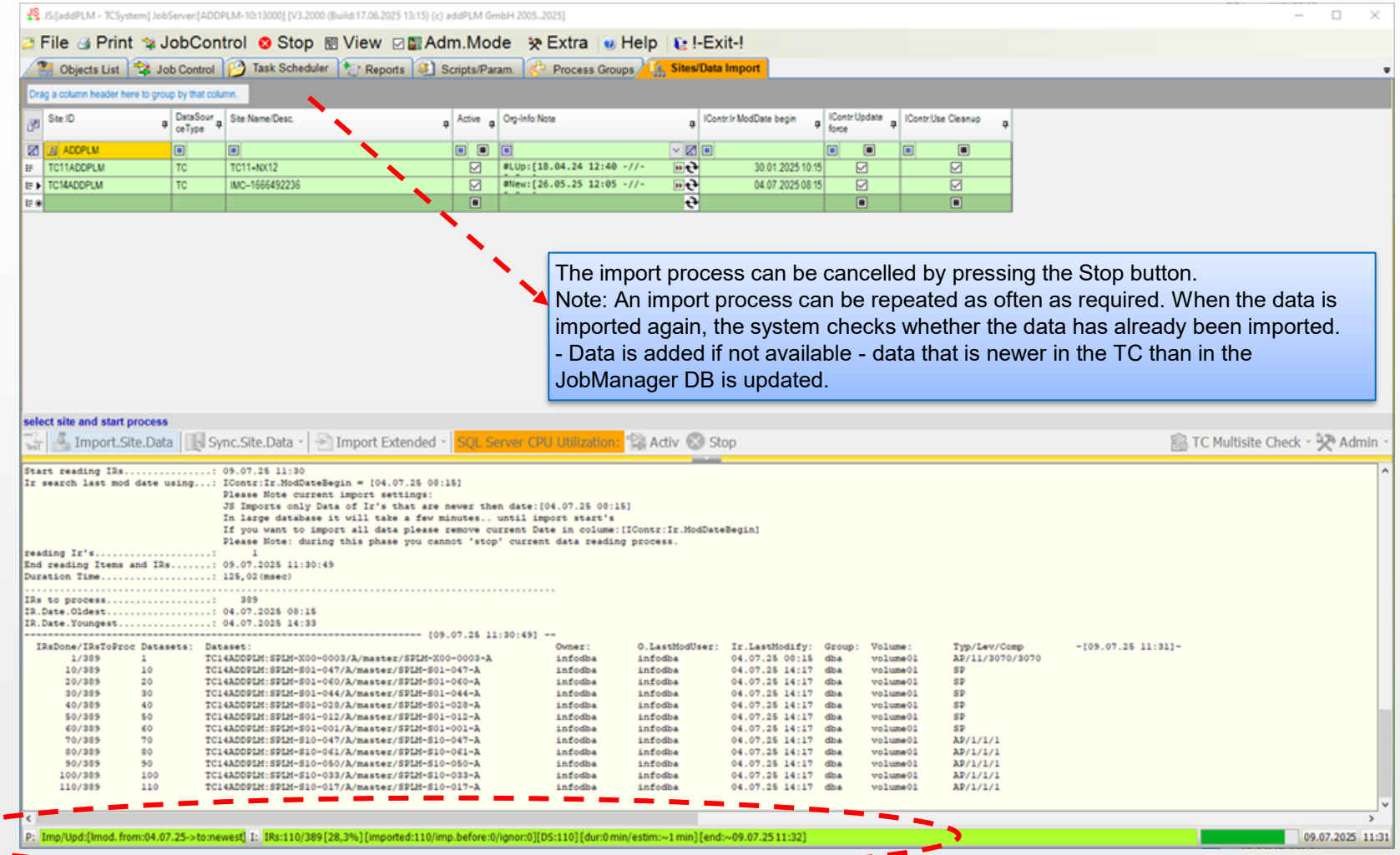
3 - Output the import information

4 - Output information about the expected import duration in the status bar.

Note: At the end of the import, a summary of the data import is displayed.
The import process can be ended by pressing the Stop (5) button.

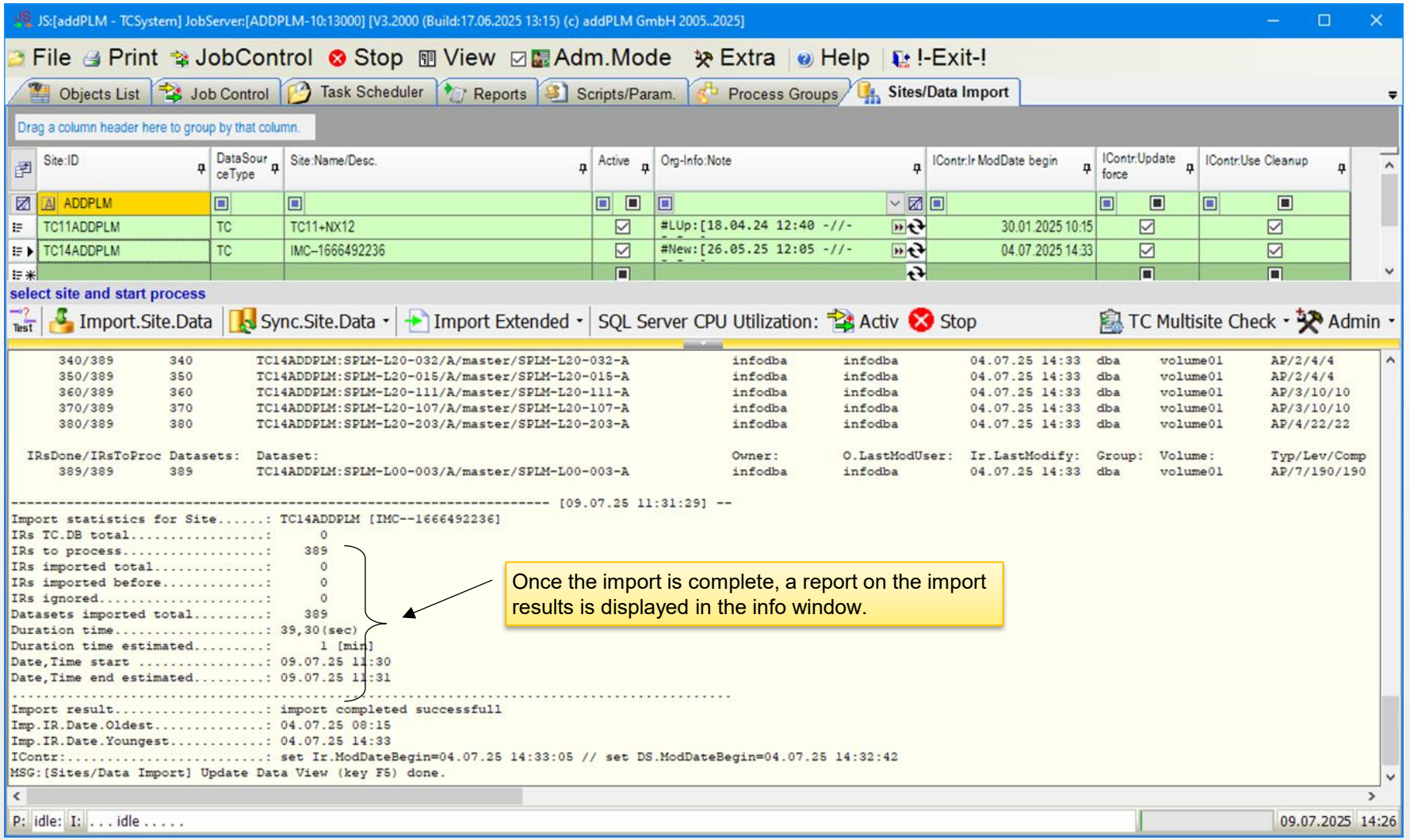
JobServer Site Data importing

- Window during the import process



JobServer Site Data importing

- Window after the import has been completed



JobServer Site Data importieren

- Settings for Native / File Based Sites

Process Description:

- JobServer do sets Searchpath for used program: '%UGII_BASE_DIR%\NXBIN\ugpc.exe‘
- Following condition is used:
  1. LoadOptions_SearchPaths: L:\\NXData\\...
     → then all sub dirs containing *.prt files will be used
  2. LoadOptions_SearchPaths: L:\\NXData\\ L:\\NXData\\NXIsoParts\\
     → then following paths are used: L:\NXData\ L:\NXData\ NXIsoParts\
  3. LoadOptions_SearchPaths: ${NXProjectPath_DP} L:\\NXData\\
     → if Env Var: NXProjectPath_DP = L:\NXData\Projekt01
     → then following paths are used: L:\NXData\Projekt01\ L:\NXData\
  4. LoadOptions_SearchPaths:
     → then no path is used ugpc.exe is using as saved Option

1.. nx120_UGII_LOAD_OPTIONS.def

```
1  LoadOptions_SearchPaths: L:\\NXData\\.
2  LoadOptions_SearchPaths: L:\\NXData\\ L:\\NXData\\NXIsoParts\\
3  LoadOptions_SearchPaths: ${NXProjectPath_DP} L:\\NXData\\
4  LoadOptions_SearchPaths:
5
6  LoadOptions_LoadComponents: load_all_components
7  LoadOptions_PartLoadOption: fully_load
8  LoadOptions_LoadWAVE: NO
9  LoadOptions_LoadWAVEParents: NONE
10 LoadOptions_LoadSubstitution: dont_allow_substitution
11 LoadOptions_LoadLatest: YES
12 LoadOptions_LoadOption: load_from_search_dirs
```

1* 2* 3* 4*

Option parameters and environment variables

- Option parameter for starting JobServer JobClient via: ..\JobManagerStart.cmd

Nr	Parameter:	Description:	Example
1	Optionsschalter: /ServerSettingsXML	Controls which serversetting file is to be read in when the software is started. Note: If this parameter is not specified, the setup file from the directory from which the NXJobManager was started is used.	/ServerSettingsXML TCSystem
2	Optionsschalter: /Setup	Starts the programme in setup mode Note No database connections are established when the software is started. The required default settings for the software can be edited here.	/setup
3	Optionsschalter: /TabsVisible #New:22.09.2022	If set, only given tabs will be visible in the JobServer If not set, all tabs will be visible in the JobServer possible values are: - ObjectList; - JobControl - TaskScheduler - Reports - Scripts - ProcessGroups - Sites	Note: multiple values are valid (separated by;) Example: /TabsVisible ObjectList;Reports;JobControl
4	Optionsschalter: /DefaultTab #New:22.09.2022	sets the given tab as the default tab, that is shown on JobServer startup possible values are: see Row above	Example: /DefaultTab JobControl
5	/DrawingIndexSearchUIShow #New:11.12.2018	JobServer if set, a special gui will be showed, only for drawing indexing search	Example: /DrawingIndexSearchUIShow
6	Umgebungsvariable: JobMgr_Root_DP JobMgr_01-BinServer_DP	Name: JobMgr_Root_DP Use : Optional Content: Root directory of the application **Note:** is overwritten by the settings in the ServerSettingsXML file Variable is set in the JobManagerStart.cmd script.	Set JobMgr_Root_DP=S:\JobManagerV3 Set JobMgr_01-BinServer_DP=S:\JobManagerV3\01-BinServer

Option parameters and environment variables

- Option parameter for starting JobServer JobClient via: ..\JobManagerStart.cmd

Nr	Parameter:	Beschreibung:	Beispiel
1	_RunAfterImportOnJobServer*.*	If files with the name pattern *RunAfterImportOnJobServer*.* is added to the package, this script or programme will be executed after the import.	Mail_MSG_RunAfterImportOnJobServer.cmd
2	Optionsschalter: /AutoStart -TaskScheduler:On -JobContol:[A,R]	This option can be used to control that the software automatically activates the specified mode after startup . Is used, for example, to start the software automatically in the desired mode after restarting the computer.	/AutoStart “TaskScheduler:On“ → startet den TaskScheduler /AutoStart “JobControl:A“ → startet JobControl im Modus:Automatic /AutoStart “JobControl:R“ → startet JobControl im Modus:Recive /AutoStart “TaskScheduler:On JobContol:A“ → startet TaskScheduler + JobControl:A
3	JobManagerStart.cmd /Client	Starts the JobServer client Note: The environment is controlled by the start script JobManagerStart.cmd	.\01-BinServer\JobManagerStart.cmd /Client
5	/IRIds_ObjectNames	Transfer of IR and object search filter for JobPlaner	/IRIds_ObjectNames 10872019/0-Gestell##UGPART##Z1;10872019/0-Gestell##UGPART##10872019-0; /IRIds_ObjectNames 000013/A##UGMASTER##000013-A;

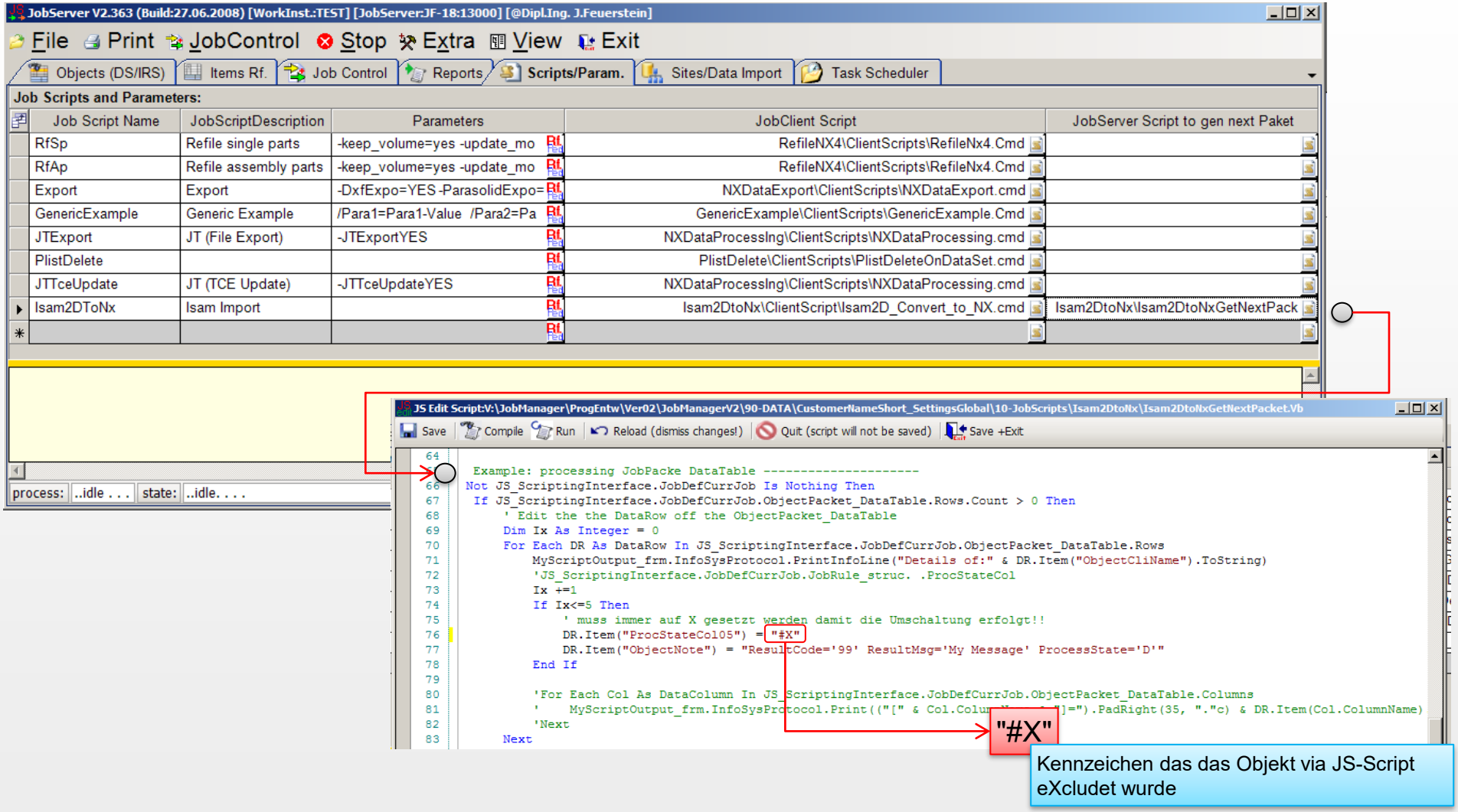
Option parameters and environment variables

- Option parameter for starting JobServer JobClient via: ..\JobManagerStart.cmd

Nr	Parameter:	Beschreibung:	Beispiel:
1	Optionsschalter: /JobClientStartMode Automatic /JobClientStartMode Test /JobClientStartMode Pause	This option can be used to control that the software automatically activates the specified mode after startup.	/JobClientStartMode Automatic → startet den JobClient im Automatic Modus /JobClientStartMode Test → startet den JobClient im Test Modus /JobClientStartMode Pause → startet den JobClient im Pause Modus d.h. die Software startet verbindet sich aber nicht mit dem JobServer. Diese Option kann verwendet werden um bei Test's zu vermeiden das der JobClient Jobs neu annimmt oder Ergebnisse an den JobServer übermittelt
2	Environ Variable: JobMgr_RootL_DP	This environment variable can be used to define that the JobManager programmes are copied locally to a specified directory and started from there.	set in user or system Environment set JobMgr_RootL_DP=D:\JobMgrL set JobMgr_RootL_DP=%ProgramFiles(x86)%\JobMgrL
3	Command line option: /WorkInstTitel #CutomTitel#	Defines the displayed tile of JobServer GUI. It can be used for easer identification off JobSever's if running serveral JobServers on the same Computer.	/WorkInstTitel PLM1-NL_VLD /WorkInstTitel PLM2-US_WLT
4	Command line option: /TCPPort #PortNumber# #New:10/2014	Defines which IP Port is used for JobServer JobClient communication can be used for Start of JobServer and JobClient	/TCPPort 14001
5	/JobMgr_JCAliasName	Defines the Name that JobClient uses to identify the JobClient against the JobServer Use case is to run processes with one JobClient processing Data in different Sites	/JobMgr_JCAliasName "%ComputerName%.TCPLMP"

JobServerScripts

- Exclude process status of a process package using a JobServer script



JobServerScripts

- Details zur Steuerung des JobServers via JobServerScripts

JS_ScriptingInterface.JobDefCurrJob.JobDef.JobServerScript_RunCondition

JS_ScriptingInterface.JobDefCurrJob.JobDef.JobRule.JobControlInterface.

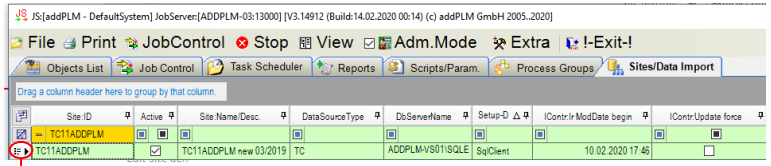
SelectObjects_ViaJobRule
JobPaket_SendToJobClient
ExcuteJobServerScript_and_ContinueWithNextJobRule

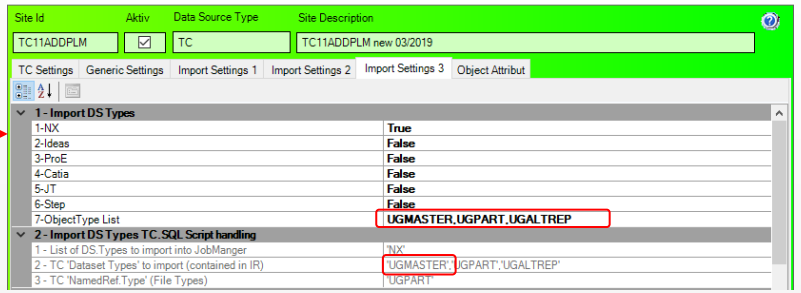
ScriptingEngineClass2 | ScriptMain

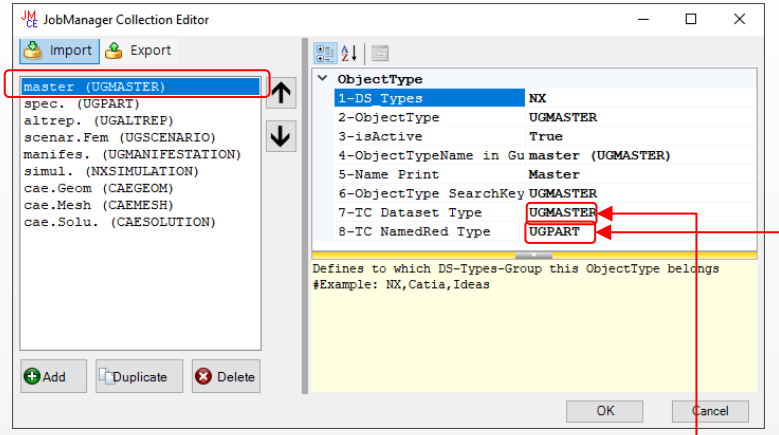
```
Select Case JS_ScriptingInterface.JobDefCurrJob.JobDef.JobServerScript_RunCondition

    Case ScriptingDef.RunCondition_enum.Cond01_GenNewJob_Bevor_SelectObjectData
        '-- Aktion vor Selection der Objekt Daten:
        '-- Vorgaben für weiteres Verhalten einstellen:Keine Objekte selectieren lassen
        JS_ScriptingInterface.JobDefCurrJob.JobDef.JobRule.JobControlInterface.SelectObjects_ViaJobRule = False
        '-- Vorgaben für weiteres Verhalten einstellen:JobPaket an den CIient senden
        JS_ScriptingInterface.JobDefCurrJob.JobDef.JobRule.JobControlInterface.JobPaket_SendToJobClient = True
        Me.ParseResultFromFile("C:\Autoexec.bat")

    Case ScriptingDef.RunCondition_enum.Cond02_GenNewJob_After_SelectObjectData
        '-- Aktion wenn Daten selectiert wurden -> Bietet die Möglchkeit die ermittelten Daten zu verwenden
        ''-- Example: processing JobPacke DataTable --------------------
        '           If JS_ScriptingInterface.JobDefCurrJob IsNot Nothing AndAlso _
        '              JS_ScriptingInterface.JobDefCurrJob.ObjectPacket_DataTable IsNot Nothing Then
        '               If JS_ScriptingInterface.JobDefCurrJob.ObjectPacket_DataTable.Rows.Count > 0 Then
        '                   ' Edit the the DataRow off the ObjectPacket_DataTable
        '                   For Each DR As DataRow In JS_ScriptingInterface.JobDefCurrJob.ObjectPacket_DataTable.Rows
        '                       MyScriptOutput_frm.InfoSysProtocol.PrintInfoLine("Details of:" & DR.Item("ObjectCliName").ToString)
        '                       For Each Col As DataColumn In JS_ScriptingInterface.JobDefCurrJob.ObjectPacket_DataTable.Columns
        '                           MyScriptOutput_frm.InfoSysProtocol.Print(("[" & Col.ColumnName & "]=").PadRight(35, ".") & DR.Item(Col.ColumnName).T
        '                       Next
        '                   Next
        '               End If
        '               MyScriptReturn.ReturnDataTable = JS_ScriptingInterface.JobDefCurrJob.ObjectPacket_DataTable
        '               MyScriptOutput_frm.InfoSysProtocol.Print("JS_ScriptingInterface.JobDefCurrJob.ObjectPacket_DataTable.Rows.Count=" & JS_ScriptingI
        '           Else
        '               MyScriptOutput_frm.InfoSysProtocol.Print("JS_ScriptingInterface.JobDefCurrJob=NOTHING")
        '           End If
        '           MyScriptOutput_frm.Pause("2- click me and script will continue")

    Case ScriptingDef.RunCondition_enum.Cond10_JobResults_Get
        '-- Aktion wenn das Ergebniss vom JobClient an den JobServer zurück übermittelt wird
End Select
```

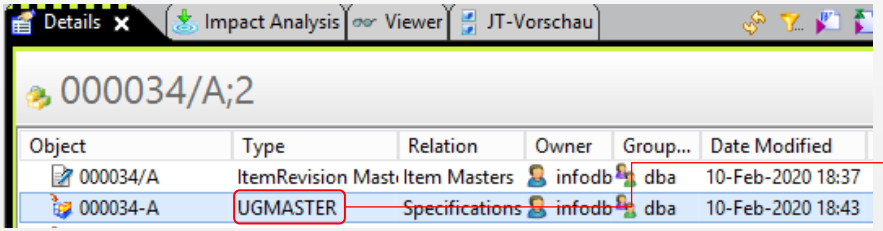
JobServer Site Definitionen

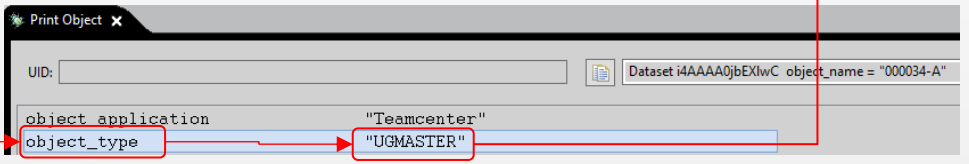
- How to define TC-Dataset Types in JobManger

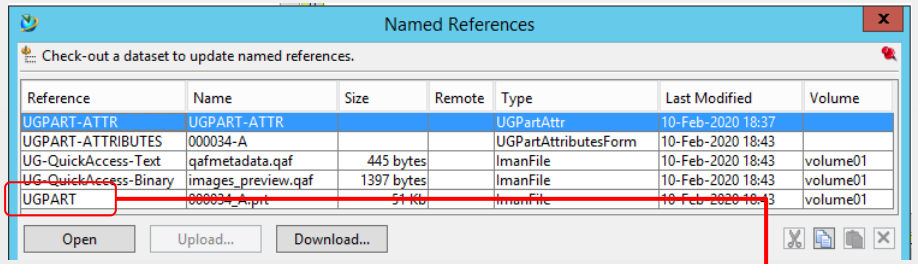


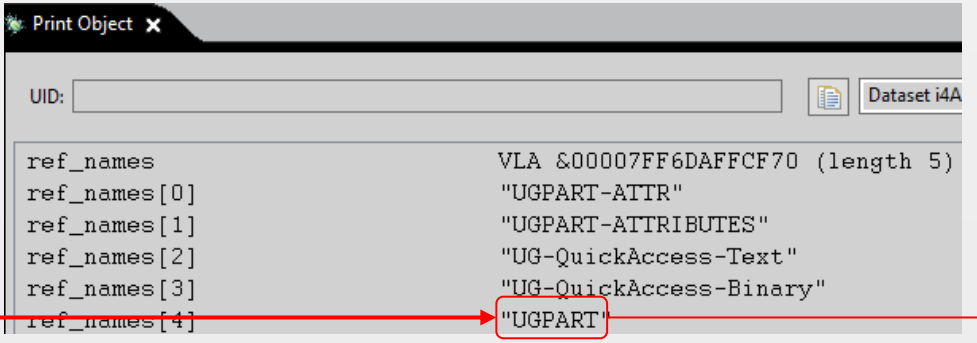












JobServer: Shortcuts

- List of keyboard shortcuts

Nr.	Situation / Abbreviation	Description:
1	JobServer is starting **Hold Shift Key**	Software is started in setup mode. After starting the software, the settings dialogue opens.
2	JobServer is starting **Hold Keys Shift + Strg**	The splash screen window displays extended information and remains open after the application is started. Any AutoStart options that have been set are not executed.
3	JobServer Data Grid View **Strg+I**	Copies the data of the active data row into the info window
4	JobServer **Strg+S**	Saves the current settings such as window positions, table views and current input data of the user interface
5	JobServer Data Grid View **Strg+R**	Adjusts the height of the table rows to the text height of the data in the table rows. This allows data row contents to be visualised very quickly

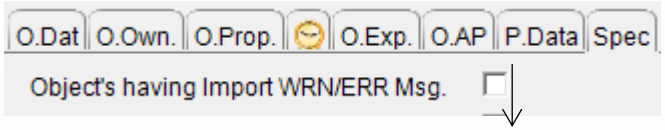
JobServer: Environ variable

- List of environment variables

Nr	Variable:	Description :	Values / Examples
1	JobMgr_Root_DP	Contains the root path in which the software is installed. The concept of the JobManager is that these are located centrally in the customer's network.	\\PLMServer\PLMShare\Prod\JobManager\.. S:\Prod\JobManager\..
2	JobMgr_ServerSetting_DP	Contains the directory in which the customer's server settings are located. %JobMgr_Root_DP%\01-BinServer.Custom	\\PLMServer\PLMShare\Prod\JobManager\01-BinServer.Custom S:\Prod\JobManager\\01-BinServer.Custom
3	JobMgr_ClientSetting_DP	Contains the directory in which the settings for the clients are located. %JobMgr_Root_DP%\02-BinClient.Custom	\\PLMServer\PLMShare\Prod\JobManager\02-BinClient.Custom S:\Prod\JobManager\\02-BinClient.Custom
4	JobMgr_RootL_DP	variable can be used to specify in which directory on a host the JobManager software should be executed.	Set JobMgr_RootL_DP=D:\PLMProgL → All programmes are then executed in D:\PLMProgL\JobManager\..
5			

JobServer: Import Analyse

- Lists all Warnings and Errors that are analysed during JobMgr Metadata Import



Key	New	Descrption:	Example
WRN_100_PSEAnaylse_LevMax			
WRN_110_PSEAnaylse_CompMax			
WRN_120_PSEAnaylse_TimeMax			
WRN_140_DataSetHasMultibleIrReferences			
ERR_130_PSEAnaylse_Ap_isSelfRefererencing			
ERR_200_DateTimeIsNull			
ERR_201_DateTimeFormatWrong			
WRN_203_RelDateIrIsMissing	07.05.2018	Ir has Status but Release Date is missing	
WRN_204_RelDateWSOIsMissing	07.05.2018	Object has Status but Release Date is missing	
ERR_301_DSIsMissingFile	11.03.14	TC Dataset Named Ref is missing req. File	
ERR_302_DSHasMutiCADFiles	11.03.14	TC Dataset Named Ref has more then 1 File reference	
ERR_402_NativNxFilePropHasZeroBytes	26.05.20	Files is corrupt because file size is 0 Bytes	
WRN_411_NativNxFileBOMAnalyse	26.05.20	General Issues found during BOM Analyse	
WRN_412_NativNxFileBOMAnalyseCompNf	26.05.20	Referenced Assembly component (File) not found	

JobServer: Glossary

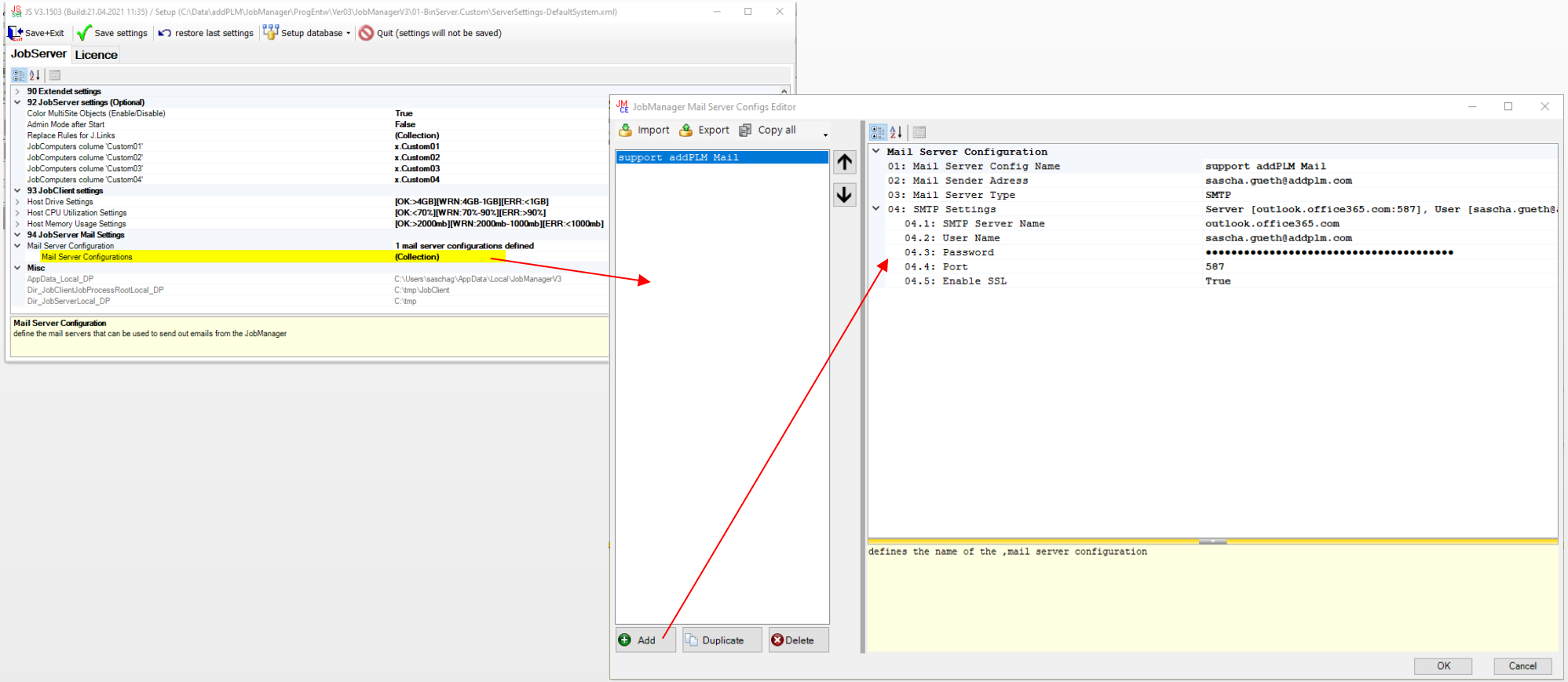
- List of Names

Nr.	Short	Description:
1	**ResCLR**	Result Classifications Rules
2	**SP**	Part.Type is Single Part / Mona-Part
3	**AP**	Part.Type is Assembly Part (IR has Bomview)
4	**SO**	Object is a Site Object and can be edit and saved on this site
5	**RO**	Object is a Replica Object
6	JS	Jobserver
7	JC	JobCLient
8	JobMgr	PLMJobManager Application

Setup JobManager EMail -> Setup Email Server

Step 1: Setup JobManager Email Server

- Define email servers that will be used to send out emails
- You can define multiple email servers
- How to define email servers:
  goto: JobServer -> Extra -> 01 – Setup – Server Settings -> 94 – JobServer Mail Settings -> Mail Server Configurations



Setup JobManager EMail -> Setup Email Configuration

Step 2: Setup JobManager Email Configuration

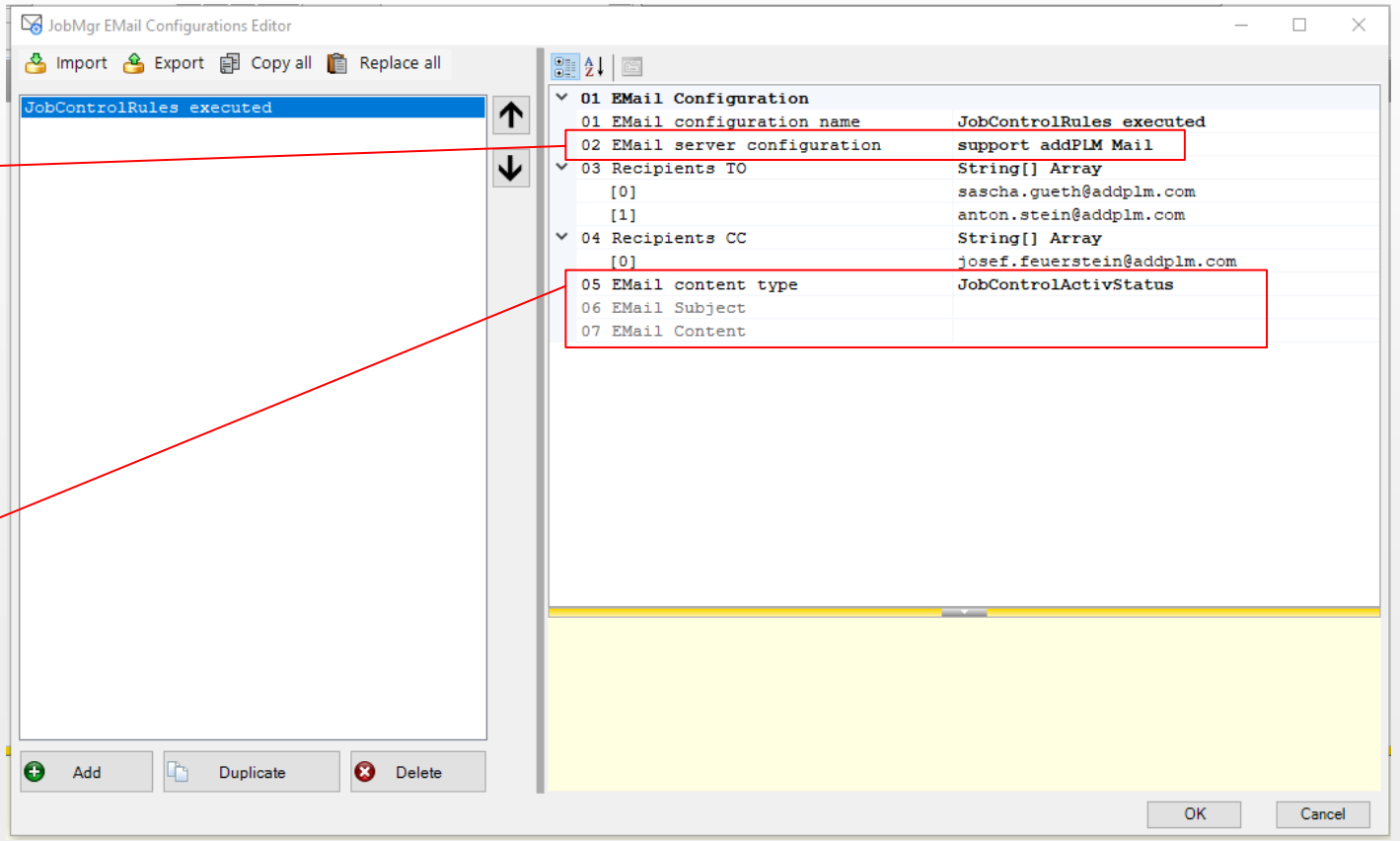
- Define email configuration that defines the email recipients and the email content
- You can define multiple email configurations
- How to define email configuration :
  goto: JobServer -> Extra -> 11 – JobManager Email Configurations

Select Email server configuration to send out the emails (defined in step 1)

Defines the EMail content type

if you select 'Custom' you can define a custom mail subject and mail content

if you select a specific type, a default subject and content will be used in the mail



Setup JobManager EMail -> Setup Email for JobControl Activ Status Rules

Step 3: Setup Email for JobControl Activ Status Rules

- Open: JobControl Activ Status Rules
  goto: JobServer -> Extra -> 09 - JobControl Activ Status Rules Edit

defines if an EMail should be send out when this JobControl Activ State Rule is executed

defines the EMail Configuration which should be used for sendig Emails
select an email configuration you have defined in step 2