

PLMJobManager V2

Anwender -Dokumentation

Erstellt von: Josef Feuerstein
 Josef Feuerstein



Inhaltsverzeichnis

<u>Aufgaben des PLMJobManager</u>	Seite: 3
<u>Das Funktionsprinzip des PLMJobManager</u>	Seite: 4
<u>Die Oberfläche des PLMJobManager</u>	Seite: 5
<u>Das Funktionsprinzip des PLMJobManager</u>	Seiten: 6-7
<u>Die Oberfläche des JobServers</u>	Seite: 8
<u>Registerkarte Objects (DS/IRS)</u>	Seiten: 9-10
<u>Such Syntax für das selektieren von Daten</u>	Seite: 11
<u>Registerkarte Objects (DS/IRS)</u>	Seite: 12
<u>JobClients Schedule – Zeiten via JobServer steuern</u>	Seiten: 13-15
<u>Definieren der ProcStateValueRule (neu 31.10.2011)</u>	Seite: 16
<u>Extra Logging – Debugging JobManager Software</u>	Seiten: 17-18



Aufgaben des *PLMJobManager*

■ Ansatz des PLMJobManager

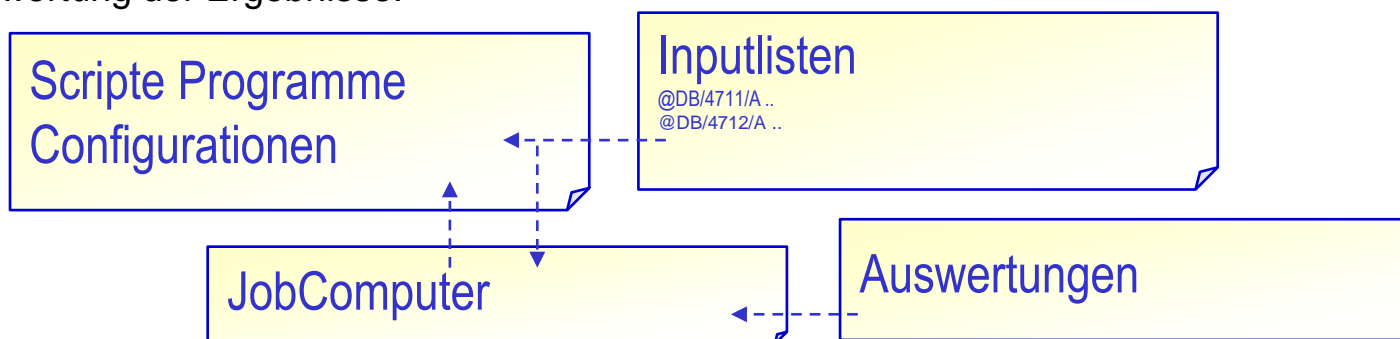
Im NX Umfeld gibt es verschiedenste Programme/Tools um Daten automatisch zu verarbeiten so .z.B.

- NX Refile
- JT-File Erzeugung via → UgToPV
- DXF-File Export via → ugto2d.exe und dxfdwg.exe
- IGES Export
- Step Export
- Parasolid Export
- NX Teilebereinigung ...

Um dieser Programme einsetzen zu können benötigt man:

- Scripte die für die verschiedenen Verarbeitungen mit den Programme deren Konfiguration steuern
- die Listen der zu verarbeitenden Objekte (Abarbeitungs- Reihenfolgen)
- Eine Steuerung der Jobs auf mehreren Rechnern
- eine Auswertung der Ergebnisse.

Schema:

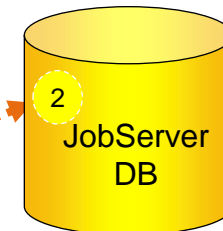
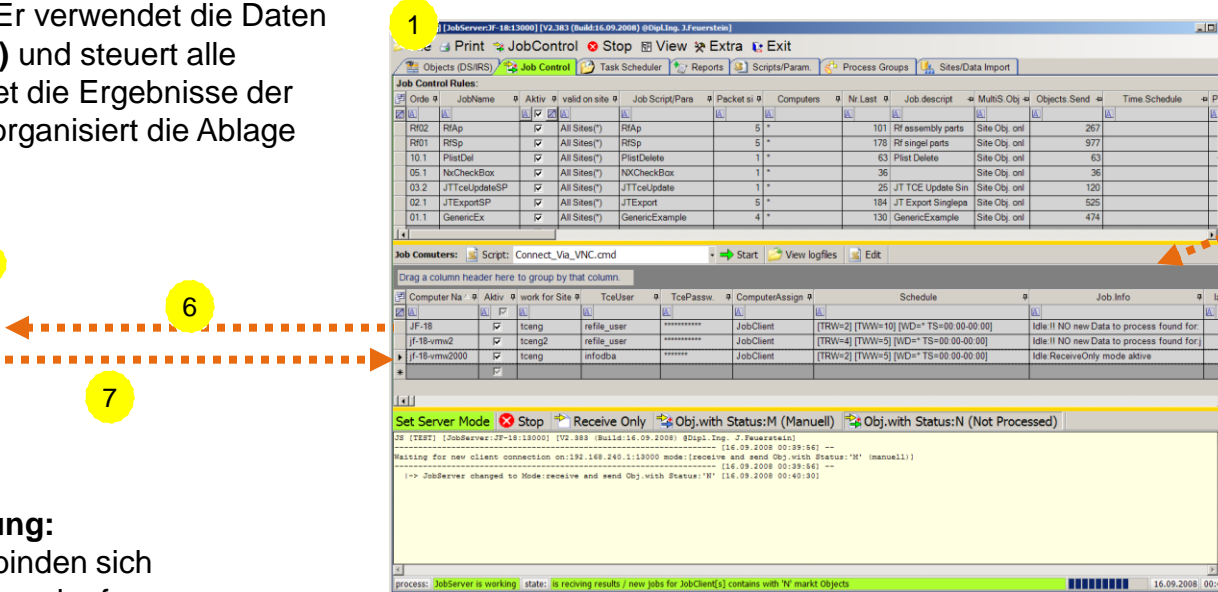


Die Aufgabe des **PLMJobManager** ist es solche Verarbeitungen einheitlich zu Organisieren und zu automatisieren.

Das Funktionsprinzip des *PLMJobManager*

■ JobContol → Übersicht wie arbeitet der '**PLMJobManager**'

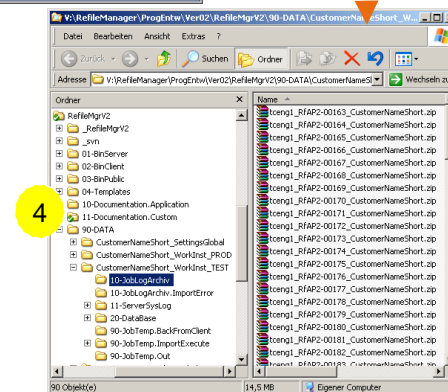
Der JobServer (1) wird zentral auf einem Arbeitsplatz installiert. Er verwendet die Daten der JobManager-DB (2) und steuert alle JobProzesse (3). Wertet die Ergebnisse der JobProzesse aus und organisiert die Ablage aller Job Logfiles (4).



Der JobServer organisiert die empfangenen JobLogfiles

Ablauf Jobverarbeitung:

Die JobClients (5) verbinden sich mit dem JobServer (1) und rufen über diesen das nächste JobPacket ab (6). Dieser Job wird nun auf dem JobClient (5) verarbeitet. Nach dem Abschluss der Verarbeitung verbindet sich der JobClient wieder mit dem JobServer und sendet die Ergebnisse an den JobServer (7). Nun ruft der JobClient wieder das nächste JobPacket ab.



Die Oberfläche des *PLMJobManager*

Die Oberfläche unterteilt sich nach den Aufgabenbereichen

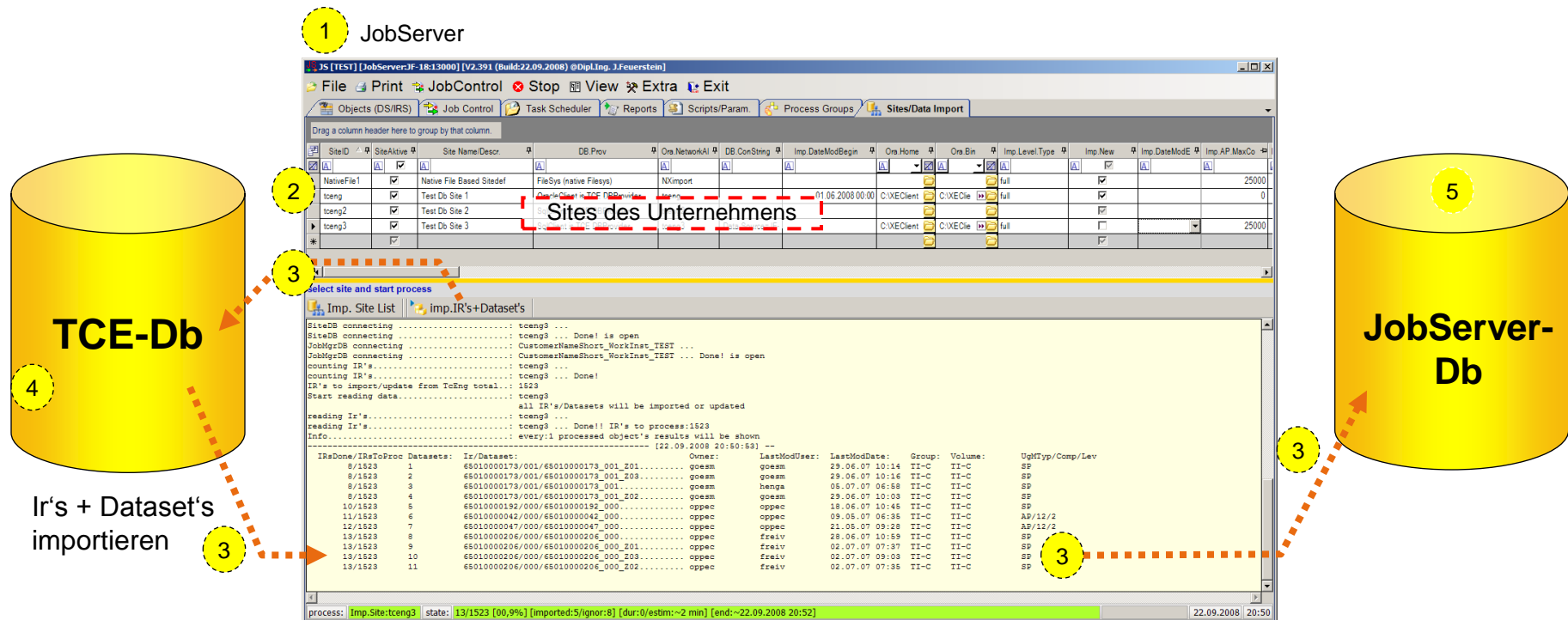
- **Objects (DS/IRS)** → Anzeige der Objekt Daten und der Refile Ergebnissen
- **Job Control** → Organisation und Kontrolle der JobClient Prozesse
- **Task Scheduler** → Steuerung von Zeitbezogenen Prozessen
- **Reports** → Auswertung der Job Ergebnisse (Listen Refile-Reports)
- **Scripts/Param.** → Erfassung und Steuerung Job Parameter und Scripte
- **Sites/Data Import** → Erfassung der Sites und Import der Daten (Item's Datasets)

The screenshot displays the PLMJobManager application window. The title bar indicates the user is 'Dipl.-Ing. J. Feuerstein' and the version is 'V2.613 (Build:02.05.2010)'. The 'Objects (DS/IRS)' tab is active, showing a list of job objects. The data grid contains 16 rows of job data, including details like 'BgStrukEx-Einzelteil-01_T1/A' and 'BgStrukEx-KBG-01_T1/A'. The bottom status bar shows the current process is 'idle' and the state is '... idle'.

Das Funktionsprinzip des *PLMJobManager*

- TCE Daten aus der TCE Datenbank in die JobServer-DB importieren.

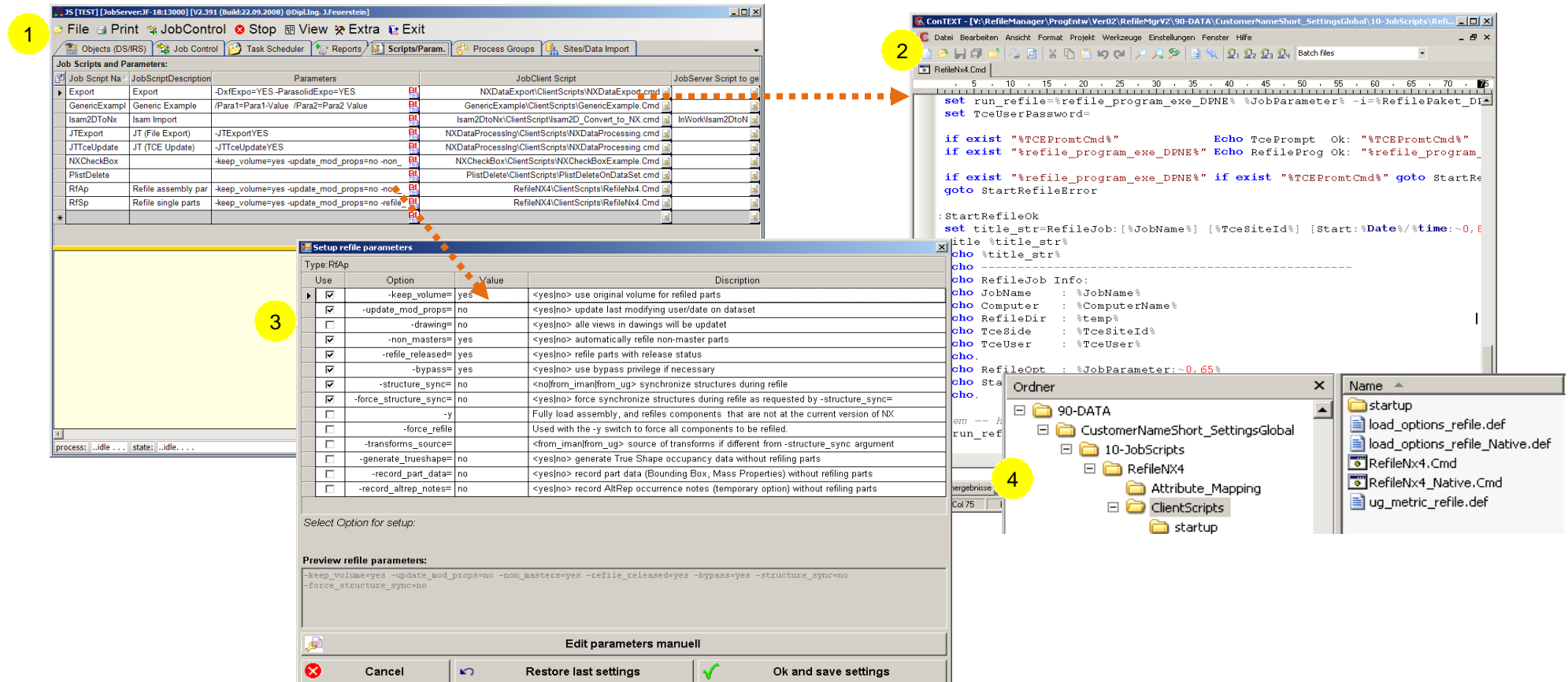
Im JobServer (1) werden die Sites des Unternehmen erfasst (2). Mit der Import/Update (3) Funktion werden aus der TCE-Db (4) die ItemRev und Datasets Daten in die JobServer-DB übertragen. Diese Daten bilden die JobServer-DB (5) und sind die Datenbasis des JobServer.



Das Funktionsprinzip des *PLMJobManager*

- Steuerung der Scripte und Parameter → Konfiguration des JobServers

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



The screenshot illustrates the PLMJobManager V2 interface with four numbered callouts:

- Job Scripts and Parameters:** A table listing various job scripts and their parameters.

Job Script Name	JobScriptDescription	Parameters	JobClient Script	JobServer Script to go
Export	Export	-DxIfExpo=YES -ParasolidExpo=YES	NXDataExport\ClientScripts\NXDataExport.cmd	
GenericExmpl	Generic Example	/Para1=Para1-Value /Para2=Para2-Value	GenericExample\ClientScripts\GenericExample.Cmd	
Isam2DToNx	Isam Import		Isam2DToNx\ClientScripts\Isam2D_Convert_to_NX.cmd	InWork\Isam2DToN
JTExport	JT (File Export)	-JTExport=YES	NXDataProcessing\ClientScripts\NXDataProcessing.cmd	
JTtceUpdate	JT (TCE Update)	-JTtceUpdate=YES	NXDataProcessing\ClientScripts\NXDataProcessing.cmd	
NXCheckBox		-keep_volume=yes -update_mod_props=no -non	NXCheckBox\ClientScripts\NXCheckBoxExample.Cmd	
PlistDelete			PlistDelete\ClientScripts\PlistDeleteOnDataSet.cmd	
RIAp	Refile assembly par	-keep_volume=yes -update_mod_props=no -no	RefileNX4\ClientScripts\RefileNX4.Cmd	
RIp	Refile single parts	-keep_volume=yes -update_mod_props=no -refile	RefileNX4\ClientScripts\RefileNX4.Cmd	
- Script Editor:** A text editor window showing a script for RefileNX4.Cmd, including commands like `set run_refile=refile_program_exe_DPNE` and `set TceUserPassword=`.
- Setup refile parameters:** A dialog box for configuring refile parameters.

Type	Option	Value	Description
<input checked="" type="checkbox"/>	-keep_volume=	yes	<yes/no> use original volume for refilled 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/for_imanfrom_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 refills 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 refilled
<input type="checkbox"/>	-transforms_source=		<from_imanfrom_ug> source of transforms if different from -structure_sync argument
<input type="checkbox"/>	-generate_trueshape=	no	<yes/no> generate True Shape occupancy data without refilling parts
<input type="checkbox"/>	-record_part_data=	no	<yes/no> record part data (Bounding Box, Mass Properties) without refilling parts
<input type="checkbox"/>	-record_altprep_notes=	no	<yes/no> record AltRep occurrence notes (temporary option) without refilling parts
- File Organization:** A file explorer window showing the directory structure, including folders like 90-DATA, CustomerNameShort_SettingsGlobal, 10-JobScripts, and RefileNX4.

Die Oberfläche des JobServers

■ Details der Oberfläche des JobServers

Die Oberfläche des JobServers ist nach den Hauptaufgaben in Registerkarten (1) aufgeteilt. Jede Registerkarte teilt sich auf nach: „Funktionsbereich (2)“ dem „Listenbereich (3)“ „dem Textausgabefenster (4)“ sowie einer „Statusleiste (5)“. Die Software wurde in Ihren Grundlagen so konzipiert, dass in allen Bereichen eine möglichst gleichartige Bedienung gewährleistet ist.

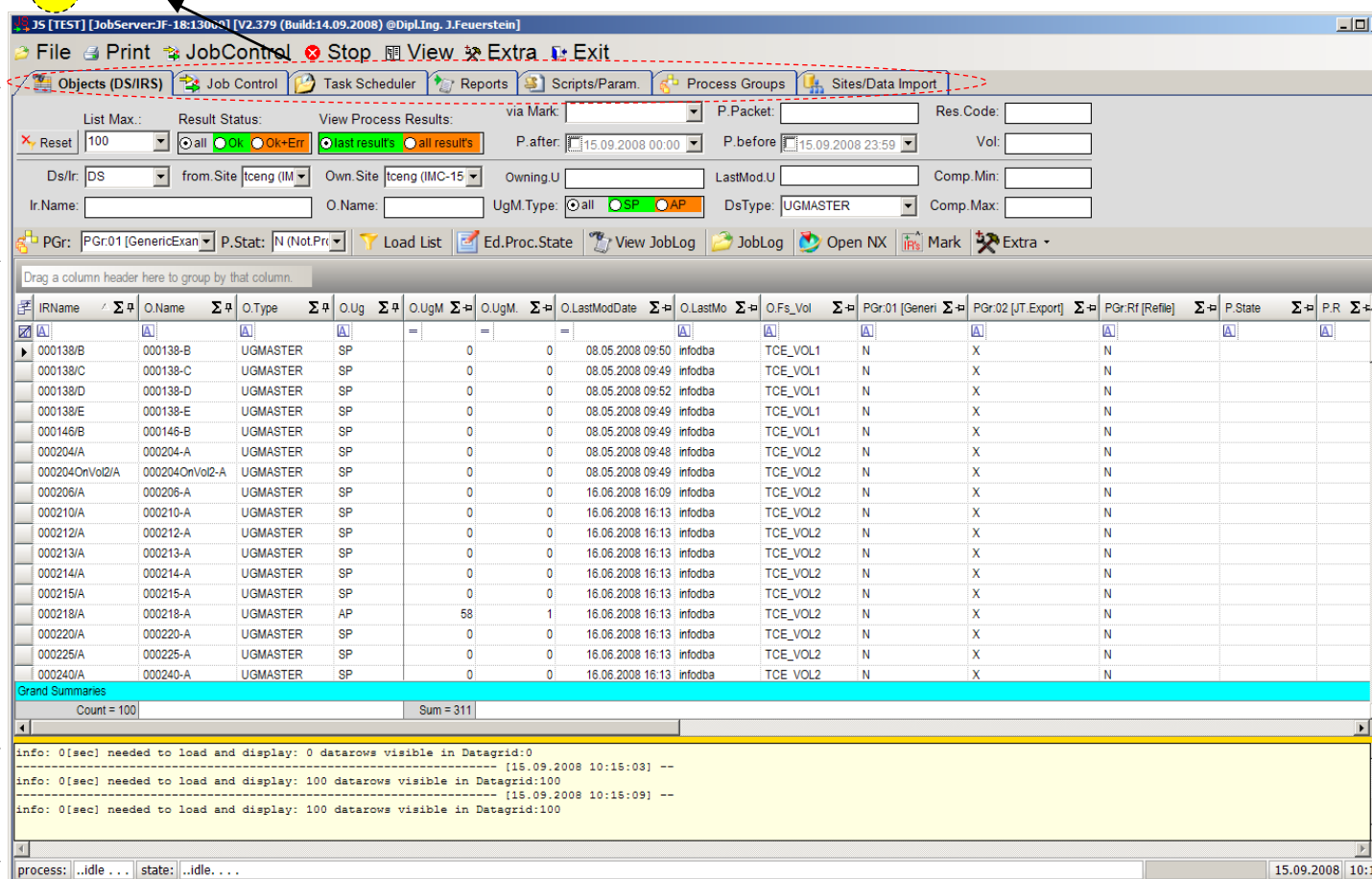
1 Registerkarten

2 Funktionsbereich

3 Listenbereich

4 Textausgabefenster

5 Statusleiste



The screenshot shows the PLMJobManager JobServer interface. The main window displays a table of job data with columns for IRName, OName, OType, O Ug, O UgM, O UgM, O LastModDate, O LastMo, O Fs_Vol, PGr:01 [Generi, PGr:02 [T.Export, PGr:Rf [Refle], P.State, and P.R. The table lists various jobs with their details. Below the table, there is a 'Grand Summaries' section showing 'Count = 100' and 'Sum = 311'. At the bottom, the 'Textausgabefenster' displays log messages, and the 'Statusleiste' shows the process state as 'idle' and the time as '15.09.2008 10:15'.

Registerkarte Objects (DS/IRS)

■ Funktions-Beschreibung der Registerkarte 'Objects(DS/IRS)'

Über diese Oberfläche können die in der Datenbank befindlichen IR/ Dataset Objekte zur Anzeige gebracht werden. Im Funktionsbereich (1) dieser Registerkarte können Filterbedingungen für die anzuzeigenden Daten angegeben werden. Die Filterbedingungen unterteilen sich nach Filterbedingungen für Prozessdaten (2) (z.B. Processed after) und nach Filterbedingungen für Objektdaten (3) und filtern nach. Die Filterbedingungen arbeiten nach dem Prinzip der Daten einschließenden Filterungen. D.h. es werden nur Daten zur Anzeige gebracht die den Filterbedingungen entsprechen.

Allgemeines Funktionsprinzip: Filter angeben und den Schalter „Load List“ (4) anwenden anschließend werden die Daten im Listebereich (5) angezeigt.

The screenshot shows the 'Objects (DS/IRS)' window in PLMJobManager. The interface includes a menu bar (File, Print, JobControl, Stop, View, Extra, Exit) and a toolbar with various icons. The main area is divided into several sections:

- 1 Funktionsbereich:** This section contains various filter and control options. It includes a 'List Max.' dropdown set to 100, 'Result Status' buttons (all, Ok, Ok+Err, Err), and 'View Process Results' buttons (last results, all results). There are also fields for 'P. Packet', 'Res. Code', 'Vol.', 'Ds/IR', 'from Site', 'Own Site', 'Owning.U', 'LastMod.U', 'Comp. Min.', 'Comp. Max.', 'Ir.Name', 'O.Name', 'UgM.Type', 'DsType', 'PGr', 'P.Stat', and a 'Load List' button (4).
- 2 Eingabebereich für Prozessdatenfilter:** This section includes fields for 'P. after' and 'P. before' with date pickers.
- 3 Eingabebereich für Objektdatenfilter:** This section includes fields for 'Ds/IR', 'from Site', 'Own Site', 'Owning.U', 'LastMod.U', 'Comp. Min.', 'Comp. Max.', 'Ir.Name', 'O.Name', 'UgM.Type', 'DsType', 'PGr', and 'P.Stat'.
- 4 Load List:** A button used to apply the filters and load the data into the list.
- 5 Listebereich:** A table displaying the loaded data. The table has columns for IRName, O.Name, O.Type, O.Ug, O.UgM, O.UgM, O.LastModDate, O.LastModUser, O.Fs_Vol, PGr.01, and PGr.02. The table shows several rows of data, including '000002/A', '000008/A', '000009/A', and various 'BgStrukEx' entries.

At the bottom of the window, there is a status bar showing 'process: ..idle ... state: ..idle ...' and a timestamp '23.09.2008 06:35'.

Registerkarte Objects (DS/IRS)

■ Übersicht über die 'Filterfelder'

Setzt alle Filter auf Standard Werte

Begrenzt auf max. zu ladende Obj.

Filter nach Result Status
 all = Status wird ignoriert (Standard)
 Ok = nur Obj. mit ResultCode=0
 Ok*Err = nur Obj. die einen ResCode haben
 Err = nur Obj. mit ResultCode<>0

Schaltet um zwischen filtern der Objektprozess Daten
 - nur letztes Ergebnis
 - aller Ergebnisse des

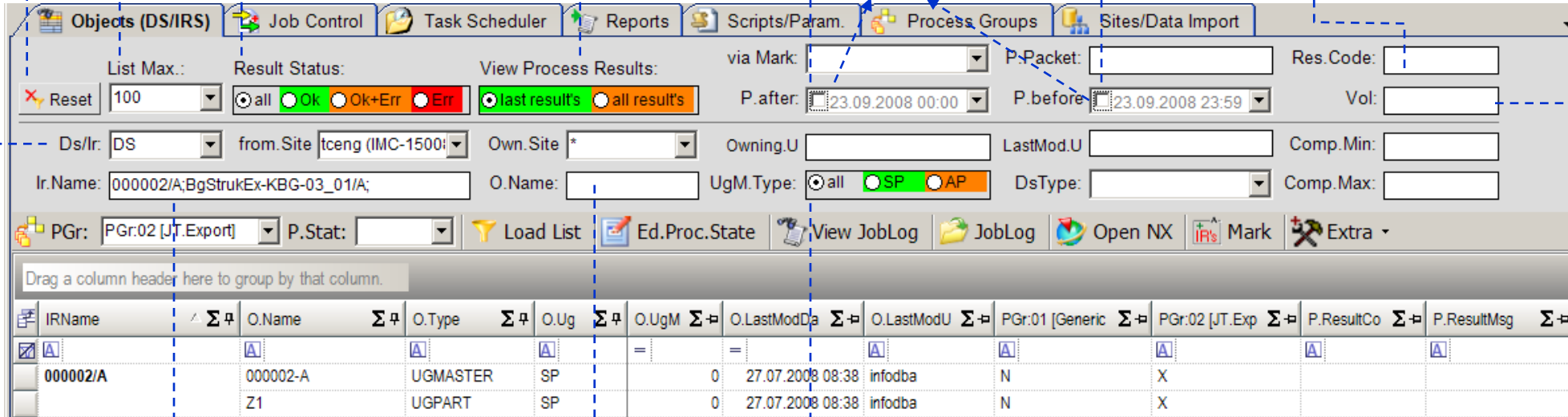
Setzt das filtern auf Objekte die **nach** dem angegebenen **Datum verarbeitet** wurden

Hinweis: das Control Kästchen setzt bzw. deaktiviert diesen Filter

Setzt das filtern auf Objekte die **vor** dem angegebenen **Datum verarbeitet** wurden

Setzt das filtern nach Res. Code z.B. 970002

Setzt das filtern nach Obj. Tce Volume z.B. TcVol01



Objects (DS/IRS) Job Control Task Scheduler Reports Scripts/Param. Process Groups Sites/Data Import

List Max.: 100 Result Status: ☒ all ☐ Ok ☐ Ok+Err ☐ Err View Process Results: ☒ last result's ☐ all result's

via Mark: P.Packet: Res.Code:

P.after: 23.09.2008 00:00 P.before: 23.09.2008 23:59 Vol:

Ds/Ir: DS from.Site: tceng (IMC-1500) Own.Site: * Owning.U: LastMod.U: Comp.Min:

Ir.Name: 000002/A;BgStrukEx-KBG-03_01/A; O.Name: UgM.Type: ☒ all ☐ SP ☐ AP DsType: Comp.Max:

PGr: PGr:02 [JT.Export] P.Stat: Load List Ed.Proc.State View JobLog JobLog Open NX IR's Mark Extra

Drag a column header here to group by that column.

IRName	O.Name	O.Type	O.Ug	O.UgM	O.LastModDa	O.LastModU	PGr:01 [Generic]	PGr:02 [JT.Exp]	P.ResultCo	P.ResultMsg
000002/A	000002-A	UGMASTER	SP	0	27.07.2008 08:38	infodba	N	X		
	Z1	UGPART	SP	0	27.07.2008 08:38	infodba	N	X		

Setzt Obj.Filter auf Ir = Filtern auf ItemRev.
DS = Dataset

Setzt Obj.Filter auf Ir Name
 Bspl: 4711 → Anzeige aller DS des Ir's 4711*
 Bspl: 4711/A; Anzeige alle DS von Ir 4711/A
 Bspl: 4711/A;00234/A; alle DS von Ir 4711/A und 00234/A
 Bspl: 4711/*;00234/*; alle DS von Ir 4711/* und 00234/*

Setzt Obj.Filter auf den Namen des Objektes
 Bspl: Z1 → Anzeige aller DS die im Namen Z1* haben

Setzt Obj.Filter nach
 SP = Single Parts = Einzelteil
 AP = Assembly Part = Baugruppe
 All = Sp + AP (Standard)

Hinweis: Das Semikolon ; wird zum trennen der IR's verwendet



Tip:

Sollten nicht die erwarteten Daten angezeigt werden so drücken Sie den Schalter „Reset“ und geben Sie die Filterbedingungen erneut ein und drücken Sie dann wieder den Schalter „Load List“. Hintergrund: In der Praxis kommt es häufiger vor das noch eine Filterbedingung mehr gesetzt ist als erwartet. Dieser wird durch den **Reset** Schalter auf Standard gesetzt so das dann mit den geänderten Filterbedingungen die zu erwartenden Daten geladen werden.



Such Syntax für das selectieren von Daten

- Dokumentation der '**Such-Syntax**' für das Auflisten oder selektieren von IR/DS Daten

'Such-Syntax'

Im JobManager gelten allgemeine Suchregeln für das Listen von Daten.

Diesen gelten für die Eingabefelder beim auflisten von DS/IRS. Wie auch beim Definieren der Datenselection für eine JobControl definition.

Hierfür wurde die folgende allgemeine Suchsyntax erstellt.

Beispiel: Listen von Object Daten

Eingabe: 4711/*;00234/*;

|→ listet alle DS von Ir 4711/* und 00234/*

Beispiel Volumes:

Eingabe: vol1;vol2; → sucht nach allen Daten die dem vol1 und vol2 gehören

Eingabe: vol1;vol2

|→ sucht nach allen Daten die genau den wert **vol1** oder **vol2** haben.

|→ Daten die andere Werte sind ausgeschlossen

Erweiterte Regeln mit Excludes

Eingabe: Vol***!not!**vol2;vol3;volLocation2*

|→ sucht nach allen Daten die mit **vol*** begingen.

|→ Daten die den Wert **vol2** oder **vol3** oder **volLocation2*** haben werden ausgeschlossen

Registerkarte Objects (DS/IRS)

Funktionsleiste Editieren des 'Prozesstatus'

Durch das Editieren des Protzesstatus können Objekte zur Prozessverarbeitung hinzugefügt werden.

1 PGr: [PGr.02 [JT.Export]]

2 Ir.Name: 000002/A;BgStrukEx-KBG-03_01/A;

3 O.Name: []

4 Edit :PGr.02 [JT.Export]

Notice: if you leave the 'from' field 'blank' target cells will be set to new value

from: X

to: M **5**

6 4 selected rows

7 X

8 M

Vorgehensweise:

- Achten sie darauf das die Richtige Prozessgruppe **(1)** ausgewählt wurde
- Selektieren die die Objekte deren Status umgesetzt werden soll **(2)**
- Klicken Sie den Schalter „Ed.Proc.Stat“ **(3)**
- Geben sie im folgenden Dialog **(4)** den gewünschten neuen Zielstatus ein Bspl. „M“ **(5)**
- Klicken Sie den Mittleren Schalter „n selected rows“ **(6)**
- Ergebnis die Editierten Objekt Prozess Status **(7) → (8)**

IRName	O.Name	O.Type	O.Ug	O.UgM	O.LastModDa	O.LastModU	PGr.01 [Generic]	PGr.02 [JT.Exp]	P.ResultCo	P.ResultMsg
000002/A	000002-A	UGMASTER	SP	0	27.07.2008 08:38	infodba	N	X		
BgStrukEx-KBG-03_01/A	BgStrukEx-KBG-03_...	UGMASTER	AP	5	16.09.2008 19:03	infodba	N	X		
	Z1	UGPART	AP	5	16.09.2008 19:03	infodba	N	X		
	Z2	UGPART	AP	5	16.09.2008 19:03	infodba	N	X		
	Z3	UGPART	AP	5	16.09.2008 19:03	infodba	N	X		
	Z4	UGPART	AP	5	16.09.2008 19:03	infodba	N	X		
	Z5	UGPART	AP	5	16.09.2008 19:03	infodba	N	X		

JobClients 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)

JobControl Rules:

J. Order	J. Aktiv	J. JobName	J. Script/Para	J. on Comp	P. Stat(N,M,A)	J. Pac	J. Nr.	J. Su	P. PGr	sel. O: (DS/IR)	sel. O: (SP/AP)
Rf90	<input type="checkbox"/>	RfSpViaDataSets	RfNx4Sp	*		10	96	729	PGr.Rf [Refile]	DS (TC DataSets)	SP (Single parts)
Rf91	<input type="checkbox"/>	RfApViaDataSets	RfNx4Ap	*		2	82	196	PGr.Rf [Refile]	DS (TC DataSets)	AP (Assembly parts)
RfNx4-1	<input checked="" type="checkbox"/>	RfNx4Sp	RfNx4Sp	*	M (Manuell sele	5	722	2127	PGr.Rf [Refile]	DS (TC DataSets)	SP (Single parts)
RfNx4-2	<input checked="" type="checkbox"/>	RfNx4Ap	RfNx4Sp	*	M (Manuell sele	1	136	373	PGr.Rf [Refile]	DS (TC DataSets)	AP (Assembly parts)

Job Computers: Script: select Start View logfiles Edit

Drag a column header here to group by that column.

C. Name	C. Aktiv	C. works for Si	Tc. User	Tc. Passw.	Org. Roll	Org. Notice	J. Schedule	J. last Info
JF-18	<input checked="" type="checkbox"/>	tceng	refile_user01	***	JobClient		[TRW=30] [TWW=60] [WD=1-5 TS=05:00-18:00 JobsMax=1] [WD=1-5 TS=18:00-05:00 JobsMax=1] [WD=6-7 TS=00:00-00:00 JobsMax=1] [WD=1-7 xReBoot=05:30-06:00]	P:NativeFile1_CBS4_00062_Customer (Dür:18:46:36)
jf-18-vmw2000	<input checked="" type="checkbox"/>	tceng	infodba	*****	JobClient	nur auf JF-18	[TRW=30] [TWW=60] [WD=1-5 TS=05:00-18:00 JobsMax=1] [WD=1-5 TS=18:00-05:00 JobsMax=1] [WD=6-7 TS=00:00-00:00 JobsMax=1] [WD=1-7 xReBoot=05:30-06:00]	!! NO new Job found for JobClient

JobServer Mode: Stop Receive Only Manuell Automatic

C. works for Site:
 Tc. User: refile_user
 Tc. Passw:
 J. Schedule: [TRW=30] [TWW=60]
 [WD=1-5 TS=05:00-18:00 JobsMax=1]
 [WD=1-5 TS=18:00-05:00 JobsMax=1]
 [WD=6-7 TS=00:00-00:00 JobsMax=1]
 [WD=1-7 xReBoot=05:30-06:00]
 J. last con.time.local.: 01.01.1900 00:00

process: idle: state: ... idle 12.01.2011 19:42



JobClients Schedule – Zeiten via JobServer steuern

■ Steuerung des der JobClient Schedule Zeiten

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]

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 $\text{ResourcePointsOfCurExecutingJobs} + \text{JobResourcePoints} > \text{JobsResPointsMax}$ 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.



JobClients 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
```



Definieren der ProcStateValueRule (Stand 23.11.2011)

■ 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 Job Ergebnisses: 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) Erforderlich

Der Wert von JobWarning (Boolean) 0=false 1=true verglichen ResWarn=* (true da Ignoriert) RW=1 (true has Warning) 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

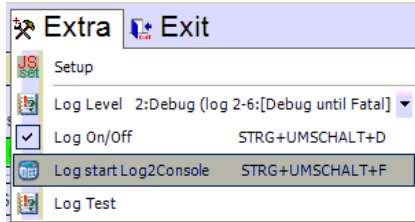
Default Values: „..\01-BinServer\Resources\ApplicationStandards\ProcStatValueRule_Template.TXT“

```
[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]
```

Extra Logging – Debugging JobManager Software

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

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

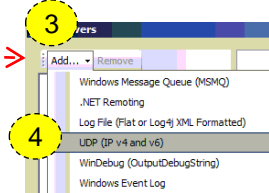
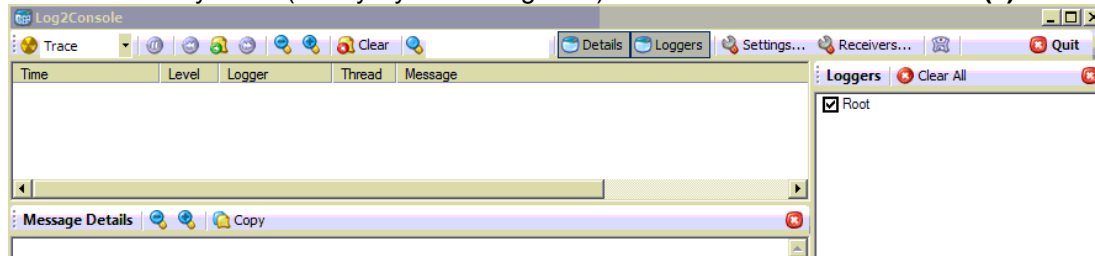


First time you use the Software on your Computer, you have to set settings in Log2Console Application:

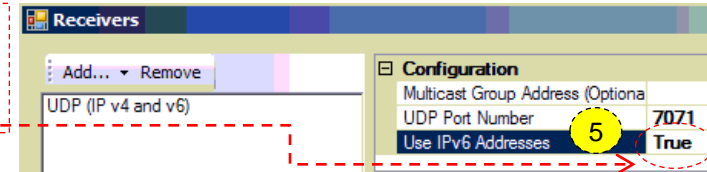
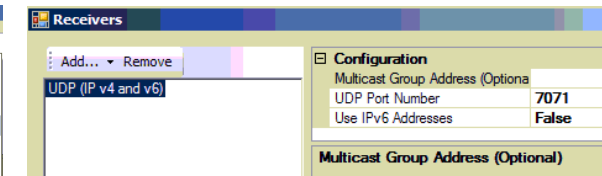


Description:

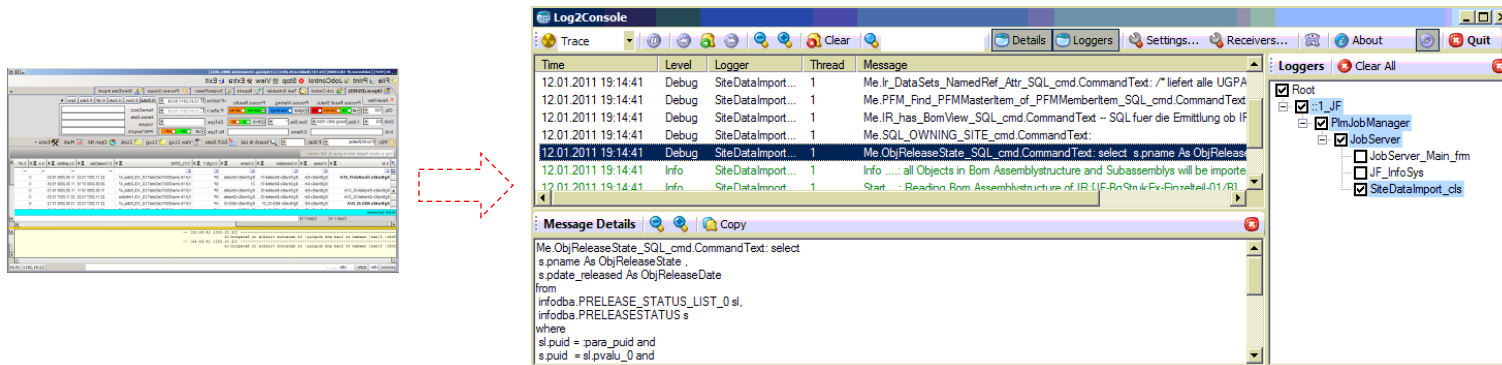
- Click Receivers (2) → Add (3) → “UDP (IP v4 and v6)” (4)
- For W7 Systems (or any System using IPv6) set “Use IPv6 Addresses” True (5)



Set on Win7
Use IPv6
Addresses to
True



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



Tastaturkürzel (Shortcuts)

■ Auflistung der Tastaturkürzel

Nr	Situation / Kürzel	Beschreibung:	Beispiel
1	JobServer startet Hold Shift Key	Software wird im Setup Modus gestartet. Nach dem Start der Software öffnet sich der Settingsdialog.	
2	JobServer startet Hold Keys Shift + Strg	Das Splashscreen Fenster gibt erweiterte Infos aus und bleibt nach dem Start der Application geöffnet. Evtl. gesetzte AutoStart Optionen werden nicht ausgeführt.	
3	JobServer Tabellenansicht Strg+I	Kopiert die Daten der aktiven Datenzeile in das Infofenster	
4	JobServer Strg+S	Speichert die Aktuellen Einstellungen wie Fensterpositionen Tabellenansichten und aktuelle Eingabedaten der Oberfläche	
5	JobServer Tabellenansicht Strg+R	Passt die Höhe der Tabellenzeilen an die Texthöhen der in den Tabellenzeilen befindlichen Daten an. Hierdurch können Datenzeilen Inhalte sehr schnell sichtbar gemacht werden	

Script Engine Change Historie

- Auflistung der Änderungen an den JobServer Klassen die im JobServer Scripting verwendet werden.

Nr	Version	Beschreibung:	Value old	Value new
1	Change: 2.901;17.11.2011;J.Fes	JobFile Definitionen	JobControlPaketGen_cls.JobFile_cls	JobFile_cls
2	Change: 2.901;17.11.2011;J.Fes	JobRule Definition geändert	JS_ScriptingInterface.JobDefCurrJob.JobDef. JobRule .JobControlInterface.JobPaket_SendToJobClient = True	JS_ScriptingInterface.JobDefCurrJob.JobDef. JobControlRule .JobControlInterface.JobPaket_SendToJobClient = True
3				
4				
5				