



HOME

Benutzer Handbuch DELMIA Process Engineer®

PoolingServer und Server Tools

The screenshot displays the IPDServer - Tools application with several active windows:

- PoolingServer Load Monitor:**

Pool Id	Machine	Process Id	Load index
0	localhost	1636	54
- Process allocated memory - NEWTONDEG [PID_1...]:**

Address	Size [kB]	Owner
00164000	816	Heap
003DF000	20	D:\DELMIA\PPRSE~1\program\bin\slrt308.dll
0040D000	48	D:\DELMIA\PPRSE~1\program\bin\IPDSE~1.E
0055A000	8	
00717000	64	VirtualAlloc/Stack/Other
00F49000	708	VirtualAlloc/Stack/Other
013CE000	4	VirtualAlloc/Stack/Other
0181E000	4	VirtualAlloc/Stack/Other
0187C000	4	VirtualAlloc/Stack/Other
- Process Monitor (ServerPool [NEWTONDEG] all processes):**

Process Name	Process Id	Processor Time [%]	Working Set [KB]	Working Set Peak [KB]	Virtual Bytes [KB]	Virtual Bytes Peak [KB]
CATSysDemon	528	0	2752	2820	21176	21176
cidaemon	1452	0	560	20656	45940	65672
clsvr	540	0	284	3028	28216	29516
CSRSS	168	0	2588	3796	19608	43424
DPFFrame	1556	0	37156	37768	144732	147384
DPFFrame	1736	0	108632	109560	236188	241268
DPFFrame	1792	0	11008	11008	37308	37308
eppoolingserver	1612	0	5904	5904	37088	37088
EPSErverTools	1760	0	10872	11256	55320	55320
explorer	1144	0	2040	11764	76972	81176
FINDFAST	1040	0	5388	64124	32448	175560
- Information:**

```

create object: EPProcessMonitor.EPProcessMonitorView.1
create object: EPSErverToolsUI.EPProcessMemoryView.1
create object: EPSErverToolsUI.EPProcessMemoryView.1
create object: EPProcessMonitor.EPProcessMonitorView.1
create object: EPSErverToolsUI.EPProcessMemoryView.1

```

EPUnlockTool



Vorwort

Das vorliegende Handbuch führt Sie in die allgemeine Bedienung und Funktionsweise der Server Tools ein.

Bei der Entwicklung der Funktionen haben wir großen Wert darauf gelegt, das Programm übersichtlich und transparent zu gestalten.

Die Bedienung und Funktionsweise wird für Sie schnell und leicht erlernbar sein - eine benutzerfreundliche Bedienoberfläche und eine übersichtliche Menüführung erleichtert Planungsaufgaben schnell und sicher im Process Engineer durchzuführen.

Trotzdem wird es noch Sachverhalte geben, die wir noch verbessern können. Sollten Sie daher Vorschläge für Verbesserungen unserer Software haben, so lassen Sie uns dies bitte wissen.

Jede konstruktive Kritik ist uns willkommen. Denn sie hilft uns, die Arbeit mit dem Process Engineer für Sie noch einfacher und übersichtlicher zu gestalten.

Dasselbe gilt selbstverständlich auch für das vorliegende Handbuch. Wenn Sie an der einen oder anderen Stelle dieses Benutzerleitfadens das Gefühl haben, dass die Funktionen oder die Programmführung nicht ausreichend erklärt werden, wenden Sie sich bitte an Ihren direkten DELMIA-Ansprechpartner. Wir freuen uns auf Ihre Anmerkungen und Vorschläge.

Ausschluss jeder Haftung und Garantie

Unsere Programme und Handbücher wurden mit großer Sorgfalt und nach bestem Wissen und Gewissen erstellt und entsprechend im Einsatz getestet. Jedoch wird keinerlei Haftung oder Gewähr dafür übernommen, dass die Software und die Beschreibungen fehlerfrei oder für spezielle Zwecke geeignet sind.

DELMIA übernimmt keine Haftung für sich aus der Verwendung dieser Software eventuell ergebende Schäden. Mit der Verwendung der Software erkennt der Benutzer diesen Haftungsausschluss an und stellt DELMIA von sämtlichen Ansprüchen frei.

Urheberrecht

Alle in unseren Unterlagen enthaltenen Informationen dürfen für interne Zwecke gerne kopiert und weiter verwendet werden, solange dies kostenlos geschieht und die Inhalte nicht verändert oder verfälscht werden.

Jede andere Form der Nutzung, insbesondere der Vertrieb auf CD- ROM oder in anderen Publikationen, insgesamt oder in Teilen, ist nur nach vorheriger schriftlicher Zustimmung durch DELMIA zulässig.

Teile dieser Software sind Eigentum der Unigraphics Solutions Inc. und urheberrechtlich geschützt. © 2002. Alle Rechte vorbehalten.

Teile dieser Software sind Eigentum der combit® GmbH und urheberrechtlich geschützt. Report-/ Druckmodul List & Label® Version 8.0: Copyright combit® GmbH 1991-2001.

Änderungen

Darüber hinaus behält sich DELMIA das Recht von Änderungen und Verbesserungen des in diesem Handbuch beschriebenen Produkts zu jeder Zeit und ohne Ankündigung vor.

DELMIA und das 3DS Logo sind eingetragene Warenzeichen von Dassault Systèmes oder Ihren Tochtergesellschaften in den Vereinigten Staaten oder in anderen Ländern.

Copyright © Dassault Systèmes 2001, 2007

Inhaltsverzeichnis

PoolingServer und Server Tools	1
Vorwort	2
Inhaltsverzeichnis	4
Einleitung	7
Wie setzen Sie dieses Handbuch ein?	7
Neue Funktionen in den Server Tools	9
Der PoolingServer	10
Überblick über die Multiserver-Umgebung	10
Serverpooling	11
Server-Prozesse, Clients und Server-Tools	13
Welche Auswirkung hat der PoolingServer auf das Setup?	15
Installation Master Server	15
Installation Slave Server	15
Registrierungs- Editor (nur bei Slave Servern)	16
Installation Client	16
Die PoolingServer-Konfiguration	16
Beschreibung der Konfigurationsdatei	16
Administrationsclients	19
Änderungen in der PoolingServer Konfiguration ab PE 5.17	20
Erweiterung des ServerPooling für explizite Administrationsclients	20
Ändern der Konfigurationsdatei im laufenden Betrieb	22
Starten und Beenden der Server Tools	24
PoolingServer und Server Tools anmelden	24
Die Menüleiste und die Werkzeugleisten	27
Das Connection Menü	27
Das <u>V</u> iew Menü und das <u>W</u> indows Menü	27
Das <u>P</u> rocess monitor Menü	27
Das <u>H</u> elp Menü	27
Arbeiten mit dem PoolingServer	28
Das Kontextmenü des PoolingServers	29
Settings...	29
Konfigurierbare Lastkriterien des PoolingServer	30
Wann sollten die Kriterien geändert werden?	33
Show ServerPool load indexes...	34
Lock PoolingServer	35
Terminate unused Manufacturing Hub server processes...	35

Terminate processes...	36
Terminate clients...	36
Reload oder Show running ServerPools	36
Abschalten der Lastmessung von PPR-Servermaschinen	37
Nicht aufgelisteten Clients	38
Das Kontextmenü des ServerPools	38
Serverpool-Einstellungen (Settings...)	38
Serverprozess Lastwerte	41
Show Process Information...	43
Terminate unused IPD server processes...	43
Terminate IPD server processes...	44
Terminate clients...	44
Reload	44
Die Server Tools	45
Arbeiten mit den Server Tools	47
Das Kontextmenü des Servers	49
Server Einstellungen (Settings...)	49
Terminate IPD server process...	51
Terminate clients...	51
Free Process Memory	51
Track Process Memory	51
Show clients oder Reload	52
Registrierungseditor Einträge	53
Zusätzlicher Registrierungseditor Eintrag: EPLOGGER	53
Optionale Registrierungseinträge	55
Das Kontextmenü auf dem Client	57
Terminate client	57
Commit Transaktion	57
Beenden von Clients	58
Vorgehensweise	58
Wo können Clients beendet werden?	58
Was geschieht beim Beenden von Clients?	58
EPUnlockTool	62
Allgemein	62
Was ist das EPUnlockTool?	62
Funktionsweise	62
Bedienungsanleitung	64
Freigabe von persistenten Sperrungen durch einen Slave Server	65
Ausgangssituation	65
Voraussetzungen und Einschränkungen	65

Abbildungsverzeichnis	66
Index	67
Anhang	69
Begriffsdefinitionen	69
Historie: Verbesserungen im Server Pooling	72
DPE5.12	72
Server Tools Client:	72
DPE5.13	72
Server Tools Client:	72
PoolingServer:	72
DPE5.14	72
ServerTools Client:	72
PoolingServer:	73
DPE5.15	73
Server Tools Client:	73
ServerTools:	74
ServerTools/PoolingServer:	74
DPE5.16	74
Server Tools Client:	74
PoolingServer:	74
Server Tools:	75
DPE 5.16SP6	75
PoolingServer:	75
ServerTools:	76
ServerTools-Client:	76
DPE 5.R17	76
PoolingServer:	76
ServerTools-Client:	77
DPE 5.17SP2	77
PoolingServer:	77

Einleitung

Die Bedienung, Funktionsweise und Menüführung, wird Ihnen in diesem Handbuch auf einfache und verständliche Weise erklärt. Es zeigt, kurz gesagt, auf, wie Sie die Server Tools und den PoolingServer für die Planungsarbeit einsetzen.

Wie setzen Sie dieses Handbuch ein?

Aufbau des Handbuchs:

Dieses Handbuch ist bewusst knapp gehalten, damit Sie schnell die Bedienung und Funktionsweise kennen lernen.

- ❑ In Abschnitt 1 [Seite 10](#) wird ein kurzer Überblick über den **PoolingServer** und die **Multiserver-Umgebung** gezeigt.
- ❑ In Abschnitt **Server Tools** [Seite 45](#) werden die Server Tools erläutert.
- ❑ In Abschnitt **Beenden von Clients** [Seite 58](#) wird das Beenden von Clients beschrieben.
- ❑ In Abschnitt **EPUnlockTool** [Seite 62](#) wird das Entfernen von persistenten Locks beschrieben.
- ❑ Im **Anhang** [Seite 69](#) finden Sie eine Erklärung zu allen in diesem Handbuch verwendeten Begriffe. Weiterhin finden Sie im Anhang eine Zusammenstellung aller Änderungen und Verbesserungen der Server Tools ab der Version DPE5.12.

Wann sollte dieses Handbuch gelesen werden?

Dieses Benutzerhandbuch richtet sich in erster Linie an Systemadministratoren oder gleichberechtigten Mitarbeitern. Alle nachfolgend beschriebenen Funktionen müssen **nicht** jedem Anwender bekannt sein.

Bevor Sie dieses Handbuch lesen, und noch nicht mit älteren Versionen des DELMIA Process Engineers vertraut sind, sollten Sie unbedingt das Basis-handbuch, den PPR-Navigator, das Administratorenhandbuch, die Handbücher zu der Projekt- und Systembibliothek sowie den Einstellungen gelesen haben. Auch die restlichen Handbücher zu lesen wäre von Vorteil.

Lassen Sie sich führen: Verwenden Sie dazu das Inhaltsverzeichnis, die Überschriften und die Kopfzeile und folgen auch den Querverweisen, die Ihnen weitere Informationen liefern.

Nutzen Sie das Wissen, das Sie aus diesem Handbuch ziehen, für alle weiteren Schritte im Process Engineer.



Hinweis:

Denken Sie daran, zu den in diesem Handbuch beschriebenen Funktionen für den Administrator sollten Sie das Wissen aus dem Basis Handbuch hinzuziehen, in dem die allgemeine Einführung in den Process Engineer beschrieben wird.



Hier rufen Sie das Benutzer Handbuch [Allgemeine Einführung](#) auf.

Wie Sie Zeichen und Symbole lesen

Die Zeichen und Symbole, die in diesem und in allen weiteren Handbüchern verwendet werden, dienen nicht nur zur allgemeinen Verschönerung eines Handbuchs, obwohl das auch eine der Aufgaben ist, sie dienen vor allem der Benutzerführung, um Ihnen den Inhalt auf leicht verständliche Weise zu erklären. Kapitel und Kapitelabschnitte werden durch Überschriften eingeleitet. Die Überschriften haben entsprechend der Verwendung unterschiedliche Schriftgrößen.

Nachfolgend wird Ihnen die Bedeutung der Symbole erklärt:



Mit diesem Symbol werden Textstellen bezeichnet, die den Funktionsumfang beschreiben, den Sie in einem Kapitel kennen lernen werden. Es steht daher in der Regel am Anfang eines Kapitels oder Abschnitts. Zudem werden wichtige Textstellen mit diesem Zeichen hervorgehoben.



Hinweis

*Mit diesem Symbol werden Hinweise gekennzeichnet, die zu einem Thema noch zusätzliche Informationen liefern, die für das Weiterarbeiten sehr wichtig sind. Das Hinweis-Zeichen kann sowohl an einem Kapitelanfang als auch bei einer bestimmten Textstelle im Kapitel stehen. Die Texte, die mit diesem Zeichen eingeleitet werden, sind zusätzlich mit dem Wort **Hinweis** gekennzeichnet. Der Text selbst ist immer kursiv geschrieben.*



Achtung

*Mit diesem Symbol werden Sie auf Sachverhalte aufmerksam gemacht, die zu möglichen Fehler bei der Bedienung des Programms führen könnten, und die Sie daher beachten sollten. Das Achtung-Zeichen kann sowohl an einem Kapitelanfang als auch bei einer bestimmten Textstelle im Kapitel stehen. Die Texte, die mit diesem Zeichen eingeleitet werden, sind zusätzlich mit dem Wort **Achtung** gekennzeichnet. Der Text selbst ist immer kursiv geschrieben.*

Beispiel

Mit diesem Symbol werden Sie auf Beispiele aufmerksam gemacht, die einen Sachverhalt verdeutlichen.



Mit diesem Symbol werden die einzelnen Bedienschritte einer Handlungsanweisung gekennzeichnet. Mit Handlungsanweisungen werden Bedienschritte beschrieben, um beispielsweise ein Menü zu öffnen oder eine Funktion auszuführen.



Mit diesem Symbol werden Aufzählungen gekennzeichnet. Das Aufzählungssymbol kann sowohl für eine Gliederung eines Fließtextes verwendet werden sowie stichpunktartig Themenschwerpunkte aufzulisten.



Mit diesem Symbol werden Sie darauf aufmerksam gemacht, dass es zu diesem Thema noch weitere Informationen in einem anderen Handbuch gibt.

Neue Funktionen in den Server Tools

Wenn Sie bereits in früheren Versionen mit den Server Tools gearbeitet haben, sollten Sie einen Blick auf dieses Kapitel werfen.



In diesem Kapitel erhalten Sie einen schnellen Überblick über alle neuen und geänderten Funktionen, die in der Version **PE 5.20** dazu gekommen sind.

Der PoolingServer

Überblick über die Multiserver-Umgebung

Was ist eine Multiserver-Umgebung?

Wird eine Software im Multiuserbetrieb auf mehreren Client PCs betrieben, übernehmen überwiegend dedizierte Servermaschinen die Speicherung von zentralen Daten, z. B. in Datenbanken. Die serverseitig laufende Software bietet den Clients Dienste zur Datenspeicherung und Abarbeitung zentraler Funktionalität an. Die heutigen Client PCs nutzen die Dienste des Servers, können aber selbst umfangreiche Anwendungen und Daten speichern und ausführen. Anwendungen, die früher nur auf dem Großrechner liefen, werden so dezentralisiert d. h. verteilt.

Werden in einer Client/Server-Software mehrere Servermaschinen für die Bereitstellung zentraler Dienste genutzt, spricht man von einer Multiserverumgebung.

Warum Multiserver-Umgebung?

Es gibt unterschiedliche Gründe für die Verwendung mehrerer Server in einer Client-Server-Anwendung. Häufig spielt dabei die Sicherheit der Datenbereitstellung, also die permanente Verfügbarkeit von Daten, eine große Rolle. Fällt in einem Client/Server Netzwerk mit einem Server, der Server aus (wie beim Großrechner), können alle Clients nicht mehr weiterarbeiten. Um dies zu vermeiden, wird der Dienst auf mehreren Servern gleichzeitig angeboten. Fällt ein Server aus, werden Clients durch eine andere Servermaschine weiterbedient. Für die performante Bereitstellung von Datendiensten ist die Multiserverumgebung eine Alternative bzw. Ergänzung zur Verwendung von Mehrprozessormaschinen, um die Leistungsfähigkeit der Serveranwendung zu steigern.

Werden in einer Multiserverumgebung Rechnerlasten (Prozessorauslastung, Speicherverbrauch) über die Servermaschinen ausgeglichen, verbessert sich die Skalierbarkeit für den Multiuserbetrieb und die Reaktionszeit für den Benutzer deutlich. Durch den Lastenausgleich wird außerdem die effektive Verwendung von Hardware sichergestellt und Leistungsengpässe verringert.

DPE unterstützt eine Multiserverumgebung mit Lastausgleich und stellt dem Administrator mit dem **PoolingServer** ein Mittel zur Hand, die Multiserverumgebung zu konfigurieren und zu verwalten.

Serverpooling

Mit dem Serverpooling werden Manufacturing Hub-Server-Prozesse in der DPE-Multiserverumgebung über mehrere Servermaschinen verteilt verwaltet, um einem Client mit einem gering ausgelasteten PPR-Server zu verbinden.

Dabei werden die Manufacturing Hub-Server-Prozesse auf jeder Servermaschine in einem Serverpool bereitgestellt. Der Serverpool ermittelt die Lastwerte der Manufacturing Hub-Server-Prozesse und überträgt sie zum Poolingserver.

Beim Verbinden eines PPR-Clients mit dem PPR-Server, vermittelt der **PoolingServer** dem Client einen Manufacturing Hub-Server aus den Serverpools mehrerer Servermaschinen. Der PoolingServer entscheidet bei einer Clientanmeldung durch den Vergleich der Lastwerte der ServerPools, welcher Server-Rechner tatsächlich verwendet wird, um ein PPRServer zu starten. Für den Lastvergleich der Serverpools wird jeweils der höchst belastete Manufacturing Hub-Server-Prozess eines jeden Pools herangezogen. Die Entscheidung, welcher Manufacturing Hub-Serverprozess der ausgewählten Servermaschine für die Clientverbindung verwendet wird, obliegt dem Zuteilungsstrategie des Serverpools.

Der **Server Pool** der ausgewählten Servermaschine stellt dem zu verbindenden Client den „nächst verfügbaren“ Manufacturing Hub-Serverprozess zur Verfügung. Die Ermittlung des „nächst verfügbaren“ Manufacturing Hub-Server-Prozesses wird durch die ServerPool-Konfigurationswerte (Clientlimit, Memorylimit) beeinflusst. Mit diesen Grenzwerten wird festgelegt, ob ein bestehender Manufacturing Hub-Server-Prozess noch Kapazität für eine neue Clientverbindung hat oder ein neuer Manufacturing Hub-Server gestartet werden muss.

Eine nähere Erläuterung der Poolingserver- und Serverpool-Konfiguration finden Sie in den Abschnitten [„Die PoolingServer-Konfiguration“](#) und [„Serverpool-Einstellungen \(Settings...\)“](#).

Die DPE – Installation besteht aus einem Master Server, optionalen Slave Servern und den einzelnen Clients.

Der **PoolingServer** ist auf dem **Master Server** installiert und teilt nach der Anmeldung eines Clients, nach festgelegten Parametern, wie etwa der freiverfügbaren Kapazität, dem Client einen Manufacturing Hub-Server auf einem ausgewählten Server Rechner zu.

Die Slave Server ermöglichen es, die Server Installation an die Anforderungen einer Multiserver-Umgebung anzupassen.

Anforderungen an die Multiserver-Umgebung sind zum Beispiel:

- Anzahl von angemeldeten Clients
- Anzahl der gleichzeitigen Zugriffe
- Datendurchsatz je Benutzer
- ...

Auf jedem Server (Slave oder Master) läuft ein Serverpool im Servertool Prozess. Ein neuer Client wird immer über den PoolingServer (**rote Linie**) angemeldet und entsprechend der frei verfügbaren Kapazität einem Server Pool (**blaue Linie**) zugewiesen.

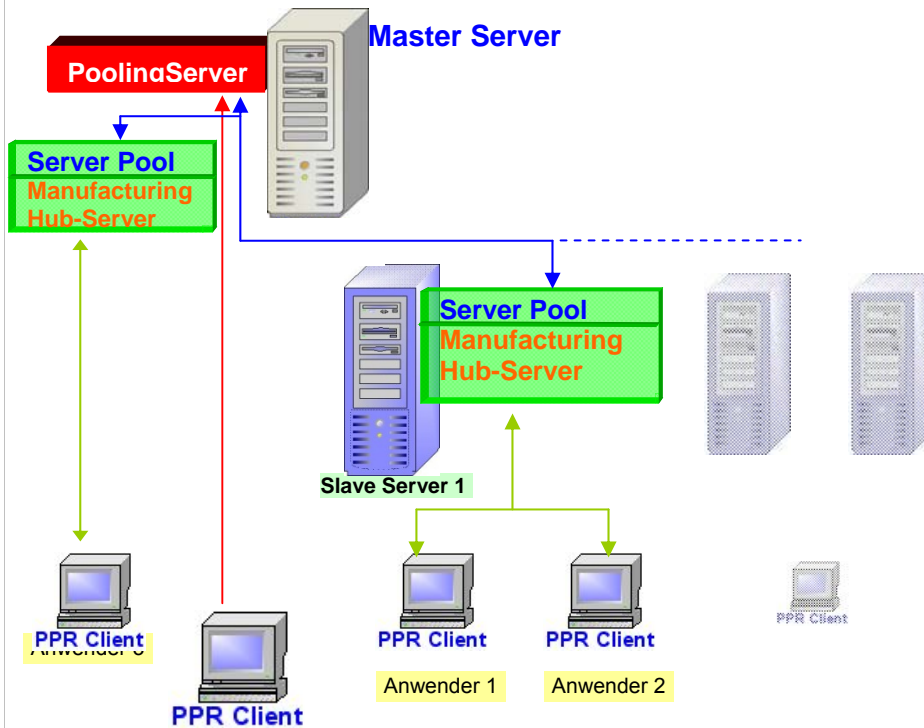


Abbildung 1: Schema PoolingServer – Server Tools und Clients

Server-Prozesse, Clients und Server-Tools

Die **Server-Prozesse** stellen dem Client Daten- Verwaltungs- und funktionale Dienste bereit.

Die Manufacturing Hub-Serverprozesse stellt dem Client den Zugang zu den Daten der Integrierten Prozess Datenbank (IPD) des DELMIA Process Engineers bereit. Daten, auf die alle Clients zugreifen, werden in der IPD zentral verwaltet. Wenn im weiteren Verlauf von einem Manufacturing Hub-Server gesprochen wird, ist immer der IPDServer gemeint.

Des Weiteren werden funktionale Servicedienste wie Datenaktualisierung, Datenspernung und Prüfung von Zugriffsrechten zur Verfügung gestellt.

Die **Server-Tools** überwachen die Manufacturing Hub-Serverprozesse und verwalten Informationen zu den angemeldeten Clients. Die Server Tools sind auf Master- und Slave -Servern installiert.

In [Abbildung 2](#) wird die Serverzuteilung an den Client über den PoolingServer schematisch dargestellt.

Kurz der Vorgang der Anmeldung eines Clients und die Zuweisung eines Servers:

- Meldet sich ein Client an, wendet er sich im ersten Schritt an den PoolingServer des Master Servers.
- Der PoolingServer seinerseits ‚kennt‘ die Daten aller Server Pools (auf den Slave Servern und/oder auf dem Master Server selbst), da er von allen Serverpools kontinuierlich Auslastungsinformationen wie CPU-Last und Speichernutzung erhält.
- Dem Client wird der Server mit der geringsten Auslastungsindex zugewiesen. Die Berechnung des Auslastungsindex und damit die Zuteilungsentscheidung kann konfiguriert und somit direkt beeinflusst werden. Auf diese Steuermechanismen wird in den nächsten Abschnitten eingegangen.



Welche Serverprozesse starten und von welchem Server sie ausgeführt werden, erfahren Sie ebenfalls in den nächsten Abschnitten.

Bevor der PoolingServer und die Server Tools näher beschrieben werden, sollte noch auf Begriffe näher eingegangen werden, die im weiteren Verlauf immer wieder verwendet werden.

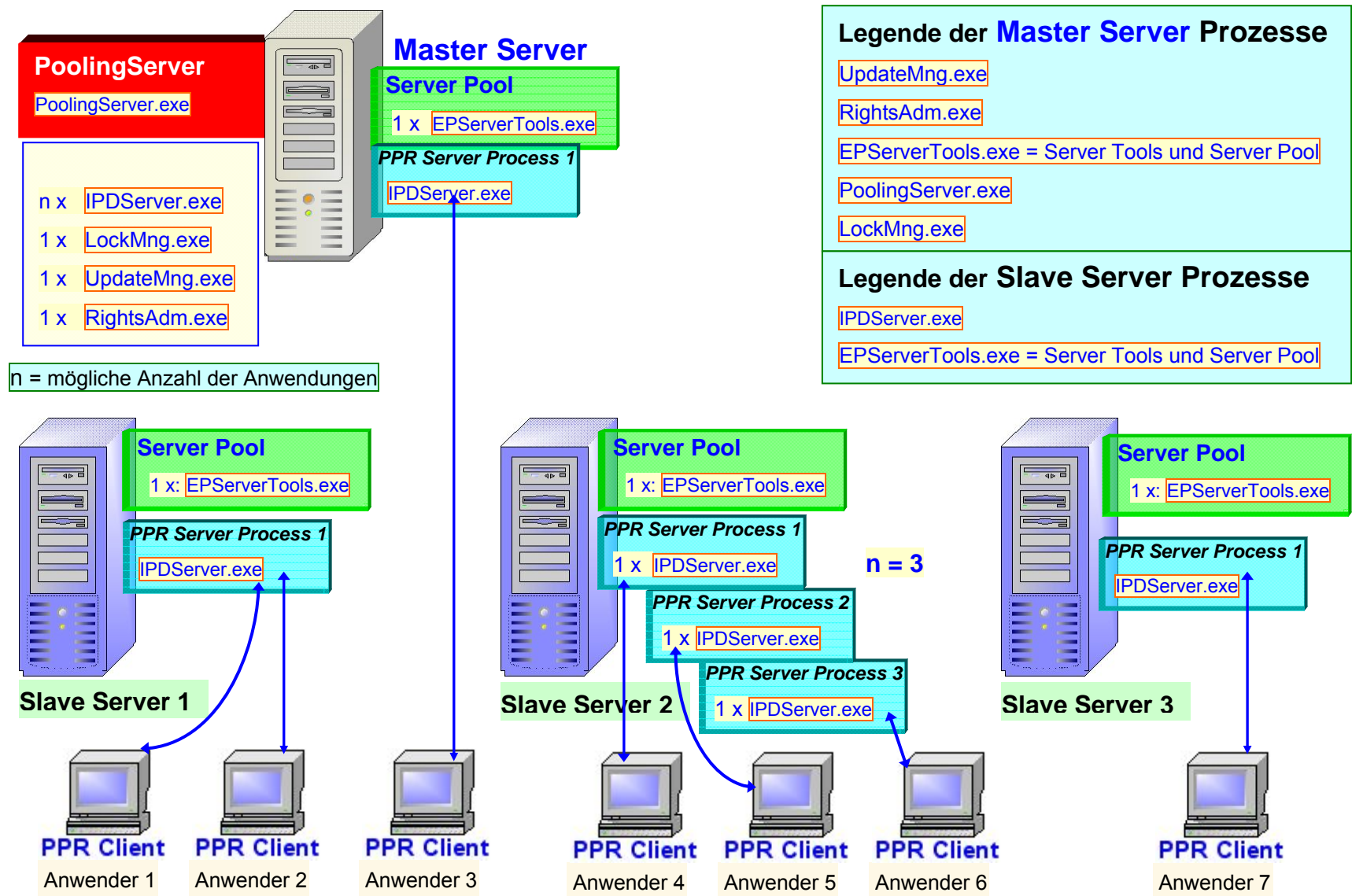


Abbildung 2: Welche Prozesse laufen wo?

Welche Auswirkung hat der PoolingServer auf das Setup?

Ab PE 5.11 wird eine Master-Slave Serverinstallation verwendet, bei der Slave-Serverrechner **nach** der Installation über die Poolingserver-Konfigurationsdatei hinzugefügt werden können (und nicht über das Clientsetup).

Dieses Hinzufügen kann zur Laufzeit des Process Engineers, also zu jedem Zeitpunkt vorgenommen werden.

Beim Clientsetup darf nur der Masterserver (Computername) angegeben werden. Auf dem Masterserver läuft ab PE 5.11 der PoolingServer, der die in der **Poolingserver-Konfigurationsdatei** aufgelisteten Serverrechner startet und bei Clientanfrage einen Server zuweist.

Installation Master Server

- ☉ Die Installation der Oracle Datenbank, des Master Servers sowie des PPR-Clients (falls gewünscht) erfolgt in gleichen Weise wie in der Version 5.10.
- ☉ Nach der Installation muss das Verzeichnis "...[Installationspfad]\PPRServer" freigegeben werden.
- ⇒ Dies wird später als gemeinsames **Data**-Verzeichnis des Servers genutzt.

	Ab PE 5.11:	Ab PE 5.12:
Installierte COM-Serverprozesse	Manufacturing Hub-Server.exe, LockMng.exe, UpdateMng.exe, RightsAdm.exe, EPSTools.exe, EPPoolingserver.exe	Manufacturing Hub-Server.exe, LockMng.exe, UpdateMng.exe, EPSTools.exe, EPPoolingserver.exe

Installation Slave Server

- ☉ Den Oracle-Service-Name eingeben.

	Ab PE 5.11:	Ab PE 5.12:
Installierte COM-Serverprozesse	Manufacturing Hub-Server.exe, LockMng.exe, UpdateMng.exe, RightsAdm.exe, EPSTools.exe	Manufacturing Hub-Server.exe, LockMng.exe, UpdateMng.exe, EPSTools.exe

Auf allen PPR Servermaschinen müssen die PPR-Serverprozesse in der DCOM-Konfiguration entsprechend der DELMIA Richtlinien konfiguriert werden. Siehe Abschnitt DCOM-Konfiguration der PPR-Serverprozesse.

Alle verwendeten PPR Servermaschinen müssen ab PE5.11 in der PoolingServers Konfigurationsdatei aufgelistet werden, damit sich PPR-Clients mit dem PPR-Server verbinden können.

Registrierungs- Editor (nur bei Slave Servern)

Die Services "LockMng", "RightsAdm", "UpdateMng" dürfen nur auf dem Master Server laufen. Ab PE 5.11 entfällt der Service "RightsAdm". Diese Registrierungs-Einträge können während der Installation nicht vorgenommen werden und müssen daher danach erfolgen. Die Einträge werden unter dem Schlüssel:

"HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\MANUFACTURING HUB-SERVER" vorgenommen.

Der Masterserver wird unter: ... \UpdateMng \ Cascaded \ Server eingetragen.

Installation Client

- ➊ Als Server den Master Server angeben
- ➋ Das freigegebenen **Data** -Verzeichnisses des Master Servers wählen
- ➌ Administration Tools installieren. Alle anderen Einstellungen belassen.

Die PoolingServer-Konfiguration

DPE Multiserverumgebung wird mit Hilfe der PoolingServer-Konfiguration eingerichtet. Werden mehrere PPR Servermaschinen verwendet, muss die PoolingServer-Konfiguration nach Ausführen des Setups bearbeitet werden, um einen Client-Server-Betrieb über mehrere Servermaschinen zu ermöglichen.

Die PoolingServer-Konfiguration wird in der Poolingserver-Konfigurationsdatei editiert und gespeichert.

Beschreibung der Konfigurationsdatei

Die Konfigurationsdatei befindet sich standardmäßig im Verzeichnis: ... \PPRServer\data\poolingsrv. In der Konfigurationsdatei finden Sie eine kurze Erläuterung zu den einzelnen Begriffen. Sie können die Konfigurationsdatei auch an einem anderen Ort speichern. Welche Einstellungen Sie dann vornehmen müssen, wird in den nächsten Abschnitten beschrieben.

Die Konfigurationsdatei enthält alle PPR Servermaschinen und optionale Angaben zur Konfiguration der Lastmessung. Die Server- und Clientmaschinen werden über Gruppennamen einander zugeordnet. Diese Zuordnung ermöglicht, dass sich eine Untermenge von PPR Clientmaschinen ausschließlich auf die zugeordnete Untermenge von PPR Servermaschinen verbindet.

Die Konfigurationsdatei kann zur Serverlaufzeit bearbeitet werden. Werden Änderungen vorgenommen, dauert es nach dem Speichern max. 30 sec bis der PoolingServer reagiert.

Nach der Installation sollte die Poolingserver-Konfigurationsdatei ähnlich wie in [Abbildung 3](#) aussehen. Hierbei wurden alle Kommentarzeilen entfernt.

Zeilen die mit
// beginnen,
sind Kommentar-
zeilen

```
/** DPE POOLINGSERVER CONFIGURATION **/  
<<netconfig>>  
server net:  
    Location [localhost(0)]  
//localhost ist standardmäßig der Master Server
```

Abbildung 3: Poolingserver-Konfigurationsdatei

Die Poolingserver-Konfigurationsdatei ist in zwei Abschnitten aufgeteilt:

dem Abschnitt zur Deklaration der Server, dem **server net** und dem Abschnitt zur Deklaration der Clients, dem **clientnet**.

- Sollen Clients nur auf bestimmte Server im Netzwerk zugreifen können, z. B. um WAN - Netzwerkverbindungen zu vermeiden, müssen mehrere Servergruppen eingerichtet werden – siehe Konfigurations-Beispiel. **b)**.
- Wenn alle Clients auf alle Servermaschinen zugreifen sollen, dann ist eine Einteilung in mehrere Gruppen nicht notwendig. Es muss lediglich eine Servergruppe angelegt werden – siehe Konfigurations-Beispiel a). Clientmaschinen müssen in diesem Fall nicht aufgelistet werden.

Nachfolgend werden zwei Möglichkeiten beschrieben:

a) Alle Clients können auf alle Server zugreifen:

```
<<netconfig>>  
servernet:  
    Fellbach [localhost, Server1, Server2]  
  
clientnet:
```

b) Es werden **Client-Server-Subnetze** definiert. So wird einer bestimmten Gruppe von Clients der Zugriff auf eine bestimmte Gruppe von Servern ermöglicht, z. B. (*localhost* ist der Master Server):

```
<<netconfig>>  
servernet:  
    Fellbach [localhost, Server1, Server2]  
    Standort2 [Server1, Server2]  
  
clientnet:  
    Fellbach [client1, client2, client5]  
    Standort2 [client3, client4, client5]
```

oder

```
<<Netconfig >>  
Servernet:  
    Standort1 [Server1 (0), Server2 (1)]  
    Standort2 [Server3 (0), Server4 (0)]  
Clientnet:  
    Standort1 [Client1, Client2]  
    Standort2 [Client3, Client2]
```

Bemerkung: - **Fellbach** , **Standort2** sind die konfigurierten Serverdomänen.

Die **Client-Server-Subnetz** Konfiguration erlaubt die Ausnutzung der lokaler Netzwerktopologie und kann zu einem besseren Datendurchsatz führen, wenn z. B. für eine Clientgruppe Verbindungen zu weit entfernten Servermaschinen mit hohen Netzwerk-Latenzzeiten vermieden werden können. Sie kann auch dazu verwendet werden, um bestimmten Clients - z. B. für lastintensive Aufgaben wie Import/Export - Servermaschinen exklusiv zuweisen.

Folgende Regeln sind zu beachten:

- es können beliebig viele Server-Domänen (Fellbach, Fellbach2) konfiguriert werden
- die Server werden innerhalb einer Server-Domäne in der Reihenfolge ihrer Leistungsfähigkeit (Maschinenpriorität) aufgelistet oder es werden die Maschinenprioritäten explizit in Klammern angeben, z.B.:
Fellbach [localhost(0), Server1(2), Server2(1)]
 - höchste Priorität ist 0 (bester Rechner)
 Die Maschinenpriorität kann für mehrere Maschinen gleich sein, z. B.:
Fellbach [localhost(0), Server1(0), Server2(0)]
- Server-Rechner dürfen in beliebig vielen Serverdomänen (Gruppen) aufgelistet werden.
 Eine Server kann also Mitglied von ein oder mehreren Serverdomänen sein, z. B.: *Standort1 [Server1]*
Standort2 [Server1]
 Wenn ein Server in mehreren Domänen aufgelistet wird, muss darauf geachtet werden, dass die Servergruppen der Netzwerktopologie entsprechen. D.h. wenn eine Servermaschine Mitglied von 2 Domänen ist, sollte die Netzwerkverbindung für Clients beider Domänen auch entsprechend verfügbar sein (Latenzzeit, Bandbreite).
- Client-Rechner dürfen in beliebig vielen Serverdomänen aufgelistet werden.
 Eine Client kann also Mitglied von ein oder mehreren Serverdomänen sein, z. B.: *Standort1 [Client2]*
Standort2 [Client2].
 Für die Zuordnung eines Client-Rechner zu mehreren Serverdomänen gilt ebenfalls die Richtlinie zur Beachtung der Netzwerktopologie, da sich der Client potentiell zu allen Server-Rechnern der zugeordneten Domänen verbinden kann, die Netzwerkkennwerte (Latenzzeit, Bandbreite) jedoch nur sehr eingeschränkt (mit dem Kriterium Serverpoolantwortzeit) im Lastbalancing-Verfahren des PoolingServers berücksichtigt werden.
- Client Rechner müssen **nicht** zwangsläufig für eine Serverdomäne aufgelistet werden. Es gibt dabei 2 Möglichkeiten:
 - a) Wenn ein Client in keiner Serverdomäne erscheint, stehen ihm **ALLE** Server zur Verfügung. Aus dieser Auswahl weist der PoolingServer beim Anmelden dem Client eine Manufacturing Hub-Serverinstanz auf dem Server Rechner zu.
 - b) Nichtaufgelistete Clients werden automatisch der reservierten Serverdomäne , @anonymous_userdomain', z. B.:
servernet:
@anonymous_userdomain [server1]
 Im Beispiel werden alle nichtaufgelistete Clients zum Server ,server1' verbunden.

- Wurde ein Client für mindestens eine Serverdomäne aufgelistet ([Fellbach](#), [Client1](#)), ist die Serverauswahl auf die in der Serverdomäne aufgelisteten Servern ([Fellbach](#) [[localhost](#), [Testserver1](#), [Testserver2](#)]) beschränkt.

Eine Minimalkonfiguration, ohne Auflistung der Clientrechner, wäre z. B.

```
<<netconfig>>
servernet:
    Standort [localhost, server1, server2]
clientnet:
```

Mit dieser Konfiguration können sich alle Clients auf die Servermaschinen ,localhost', ,server1', ,server2' verbinden.

Maschinennamen

In der PoolingServer Konfiguration können Maschinennamen wie folgt eingegeben werden:

- 1 Alphanumerische ASCII -Zeichenfolge (0...9, A...Z, a...z) ohne Leerzeichen und ohne Sonderzeichen. Zusätzlich können ein Punkt '.' und/oder ein unterstrich '_' enthalten sein.

Beispiel

```
servernet:
    [ abc123, abc_123 , abc.def ]
```

- 2 Wenn ASCII-Zeichen verwenden werden sollen, die in 1) nicht zugelassen sind, muss die Zeichenfolge in '...' (= einfache Hochkomma) geschrieben werden.

Beispiel

```
servernet :
    [ 'abc-de-123', 'abc de', 'abc:def', '#123abc' ]
```

- 3 UNICODE-Zeichen außerhalb des ASCII-Bereiches sind generell in der PoolingServer-Konfigurationsdatei nicht zugelassen.

Administrationsclients

Administrationsclients sind Clientmaschinen, die sich auch über einen gesperrten PoolingServer mit einem Manufacturing Hub-Server verbinden können. Sie werden durch Zuordnung zur admin-Domäne der PoolingServer-Konfiguration deklariert.

- ❑ Die admin-Domäne wird im Abschnitt 'clientnet' der Poolingserver-Konfigurationsdatei deklariert:

Beispiel

```
clientnet:
    @admindomain [client1, client2]
```

Die Maschinen client1 und client2 können sich über den PoolingServer mit einem Manufacturing Hub-Server verbinden, auch wenn sich der PoolingServer im Sperrmodus, z. B. durch Setzen von 'locked_on_start' = 1 oder durch manuelles Sperren in den Server Tools, befindet.

Änderungen in der PoolingServer Konfiguration ab PE 5.17

Ab der PE Version 5.17 gibt es Änderungen in der PoolingServer Konfiguration:

Das automatische Priorisierung von Maschinen in der PoolingServer Konfiguration ist geändert worden:

Vor PE 5.17:

Wenn keine Priorität für eine Maschine in der PoolingServer Konfiguration angegeben wurde, wurde die Priorität automatisch in die Reihenfolge des Maschineneintrags innerhalb der Domäne eingesetzt. Um automatische Priorisierung auszuschalten, muss manuell jede Maschinenpriorität gesetzt werden.

Ab PE 5.17:

Die automatische Priorisierung wurde dadurch ersetzt, dass jede Maschine mit der höchsten Priorität gesetzt ist, bis die Priorität manuell in die Konfigurationsakte geändert wird

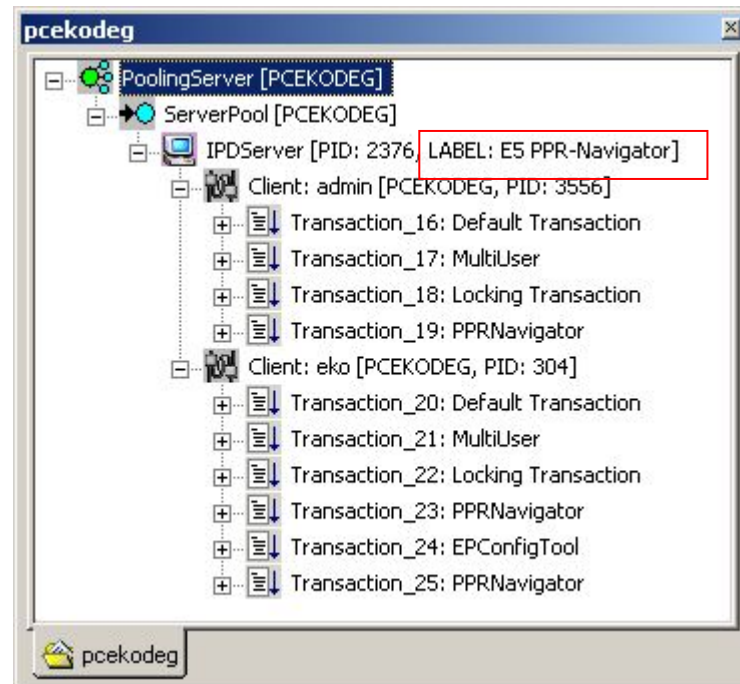
Erweiterung des ServerPooling für explizite Administrationsclients

Die Konfigurationsdatei des PoolingServer erlaubt, Client- und Servermaschinen zu gruppieren und einander zuzuordnen. Bei der Verwendung dieser Konfiguration wird ein Client mit einer Servermaschine verbunden, die entweder vordefiniert ist oder von gewissen Parametern abhängig, ausgewählt wird. In den meisten Fällen wird eine Servermaschine für mehr als einen Client konfiguriert. Je nach der aktuellen ServerPool Clientbegrenzung und der Anzahl der angeschlossenen Clients wird eine gewisse Anzahl von Manufacturing Hub-Server Prozessen zwischen den Clients gestartet und gemeinsam benutzt.

Bis einschließlich PE5.16 ist es in der PoolingServer Konfiguration nicht möglich, einem Client einem Manufacturing Hub-Server Prozess exklusiv zuzuweisen, nur die Servermaschine kann angesprochen werden. Für einige Clients, insbesondere die Administratorenclients ist es jedoch notwendig einen Manufacturing Hub-Server Prozess ausschließlich zu verwenden. Dadurch ist der Administratorenclient in der Lage die gesamten Ressourcen eines Prozesses (Speicher, CPU per Thread) zu verwenden, um beispielsweise kritische Aufgaben zu beschleunigen oder zum Speichern von Prozessen.

Ab PE5.17 wird der PoolingServer um eine neue Schnittstellenmethode erweitert. Die neue Methode unterstützt den Anschluss exklusiver Clients. Dafür wird im ServerPool ein Manufacturing Hub-Server für eine als exklusiv gekennzeichnete Clientgruppe reserviert. Diese Identifikation der Serveranwendung wird von den neuen Methoden für den IEPDClient und der IEPPoolingServer Schnittstelle geliefert.

Zusätzlich zur Fähigkeit, Manufacturing Hub-Server Prozesse für eine exklusive Belegung (Im weiteren Verlauf Label genannt) zu reservieren, müssen Clientanwendungen auch auf Prozessebene ansprechbar sein, um ein anderen ServerPooling für verschiedene Clientanwendungen auf derselben Maschine zu liefern. Wenn ein exklusives Label auf einem Manufacturing Hub-Server Prozess angegeben ist, wird es vom ServerTools Client angezeigt



Die PoolingServer Konfiguration im Detail

Clientanwendung

Zur Identifikation der Clientanwendung ist die PoolingServer Konfiguration um zwei neue Kennzeichnungen erweitert worden.

- Zum einen wird die Clientanwendung durch ihren EXE Namen identifiziert, im weiteren Verlauf als Prozess-Label bezeichnet:

`@@ <client process label>`.

- Zum anderen wird die Clientanwendung durch den Clientanwendungstyp identifiziert, weiterhin als Modul-Label bezeichnet:

`@<client module label>`.

Diese Bezeichnungen werden zusätzlich zum Clientmaschinennamen in der PoolingServer Konfiguration eingetragen.

Beispiel für das Clientnetz:

```
clientnet:
  Location [ client1@@dpfframe, client2 ]
```

//Beispiel für ein `process label`

Client1@@dpfframe und client2 gehören dem Clientdomänen- **Location**, der allen für 'Location' konfigurierten Servermaschinen zugewiesen werden.

Client1@@dpfframe wird von allen Clientprozessen deren Namen 'dpfframe.exe' ist ausgeführt.

```
clientnet:
  Location [client1@label1 ]
```

//Beispiel für ein `module label`

Serveranwendung

Die Serveranwendung kann über ihre Anschlussoptionen spezifiziert werden. Drei neue Optionen sind hinzugekommen:

- **connect_exclusive(<server exclusive label>)**
schließt an einen exklusiv gekennzeichneten Manufacturing Hub-Server Prozess an. Damit können exklusive gekennzeichnete Manufacturing Hub-Server Prozesse wieder verwendet werden.
- **connect_exclusive_unique**
schließt an einen Manufacturing Hub-Server Prozess an, der eindeutig gekennzeichnet ist. Der Manufacturing Hub-Server wird automatisch beendet, wenn kein Client mehr angeschlossen ist.
- **connect_shared**
schließt an alle konfigurierten Servermaschinen an, die nicht exklusiv verwendet werden.

```
clientnet:
  Location[
    client1@@dpfframe ( @opt = { connect_exclusive_unique } )
  ]

servernet:
  LocationAdmin          [ Server1 ]
  @anonymous_userdomain [ Server1, Server2 ]

clientnet:
  LocationAdmin
  [
    AdminClient1, AdminClient2,
    Server1@@pprloader ( @opt = { connect_exclusive_unique } )
  ] @opt = { connect_exclusive('domain admin') }
```

Ändern der Konfigurationsdatei im laufenden Betrieb

Die PoolingServer-Konfigurationsdatei kann zu jedem Zeitpunkt geändert werden, auch wenn der PoolingServers bereits läuft.

Wird die Konfigurationsdatei geändert, wenn der PoolingServer bereits läuft und ist diese Änderung fehlerhaft (z. B. Syntaxfehler, nicht existierender Computernamen usw.), wird die zuletzt fehlerfreie Konfiguration verwendet. Der Fehler wird in die PoolingServer-Logdatei ausgegeben. Wird der Fehler nicht behoben, werden fortlaufend Fehlerreferenzeinträge in die Logdatei geschrieben, da die PoolingServer-Konfiguration im Fehlerfall alle 5 Sekunden neu eingelesen wird. Nach Ändern der Konfiguration sollte daher das Protokoll im Log-Verzeichnis überprüft werden.

- ⇒ Der PoolingServer läuft in dem Fall einer fehlerhaften Änderung der Konfigurationsdatei, mit der zuletzt gültigen Konfiguration weiter. Im Fehlerfall wird die Konfiguration in einem 5 sec. Abstand eingelesen. Bei einer fehlerfreien Konfiguration beträgt das Leseintervall 30 sec. Beim wiederholten Einlesen der Konfiguration wird die neue mit der bereits verwendeten Konfiguration verglichen und Änderungen in den laufenden Betrieb übernommen.

Wenn die PoolingServer-Konfiguration vor dem Start des PoolingServers geändert wird, kann erst beim Start des PoolingServers festgestellt werden, ob die Konfiguration fehlerfrei ist. Im Fehlerfall beendet sich der PoolingServer direkt nach dem Start. Es wird eine Meldung mit der Nummer der fehlerhaften Zeile der Konfigurationsdatei und dem ungültigen Token in die Poolingserver-Logdatei geschrieben. Der PoolingServer kann nach dem Start nur mit einer gültigen Konfiguration laufen. Es muss daher die Konfigurationsdatei zuerst fehlerfrei sein.

Der PoolingServer wird automatisch gestartet, wenn sich der erste Client verbindet.

- Startet der PoolingServer, wird geprüft, ob mindestens ein Serverpool zur Verfügung steht.
- ⇒ Wenn nicht, wird der PoolingServer wieder beendet und der Client erhält eine Fehlermeldung.

Starten und Beenden der Server Tools

Die Server Tools starten Sie wie den Delmia Process Engineer entweder über das Startmenü (siehe auch: [Abbildung 4](#)) oder per Doppelklick auf das Symbol auf dem Desktop. Um das Symbol für den Process Engineer auf dem Desktop anzuzeigen, können Sie es beispielsweise über das Startmenü mit dem Desktop verknüpfen.

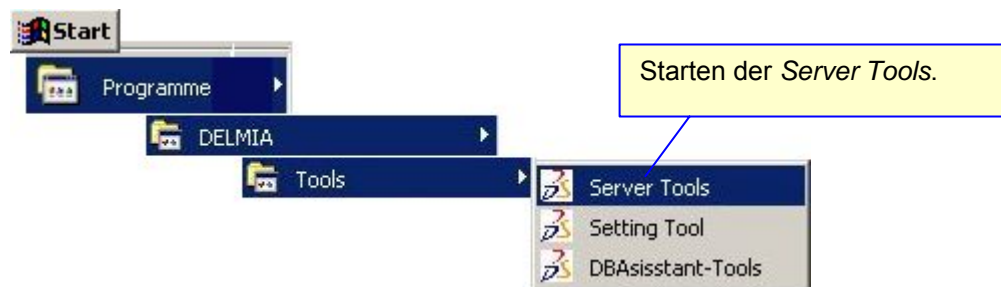


Abbildung 4: Server Tools über das Startmenü starten

PoolingServer und Server Tools anmelden

Nach dem Starten des Programms erscheint eine leere Browseroberfläche, die Oberfläche der Server Tools, wie in [Abbildung 5](#) dargestellt.

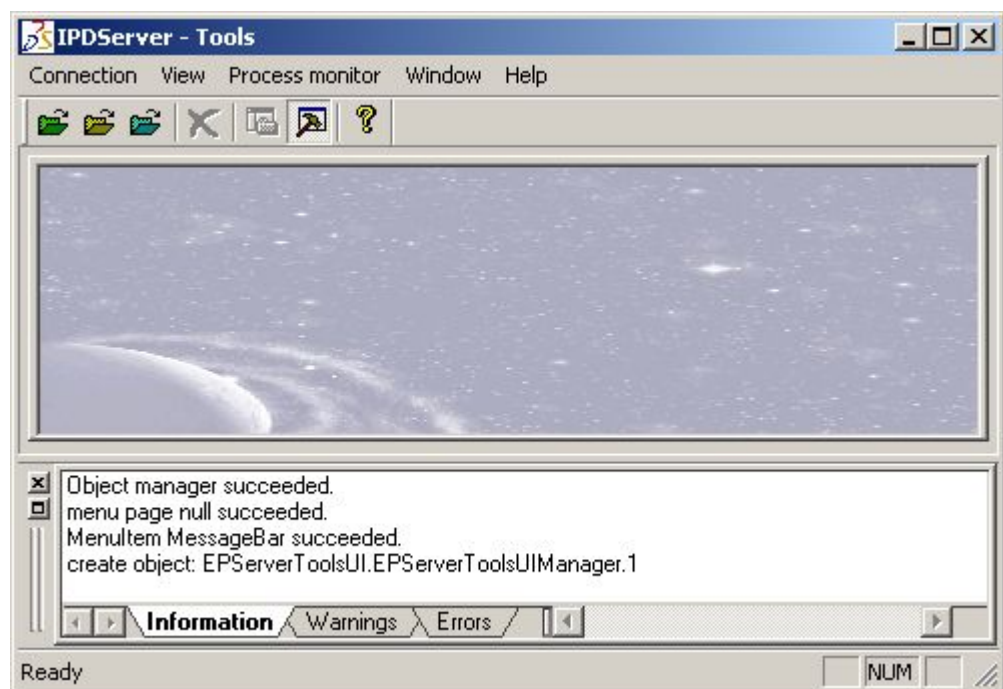



Abbildung 5: Oberfläche beim Start der Server Tools

Um sich bei dem Server anzumelden, müssen Sie in der Menüleiste *Connection* auswählen und in dem sich öffnenden Menü **Connect ...** aktivieren, oder Sie aktivieren eins der Icons in der Werkzeugleiste: .

⇒ Es öffnet sich der Dialog *Connect*.



- ☐ Mit **Connect PoolingServer** verbinden Sie sich mit dem PoolingServer, und damit mit dem Master Server, und allen damit verbundenen Slave Servern.



- ☐ Mit **Connect ServerPool** verbinden Sie sich mit den Servertools eines bestimmten Servers (in [Abbildung 6](#) mit dem gleichen Server = Master).



- ☐ Mit **Connect LockMng** verbinden Sie sich mit dem Lock-Manager. Es öffnet sich ein Dialog in dem die gelockten Clients aufgeführt werden. Diese Funktionalität steht Ihnen nur in Verbindung mit V5 zur Verfügung. Dazu müssen Sie ein Objekt in V5 geöffnet haben. Siehe auch: [Abbildung 7](#).

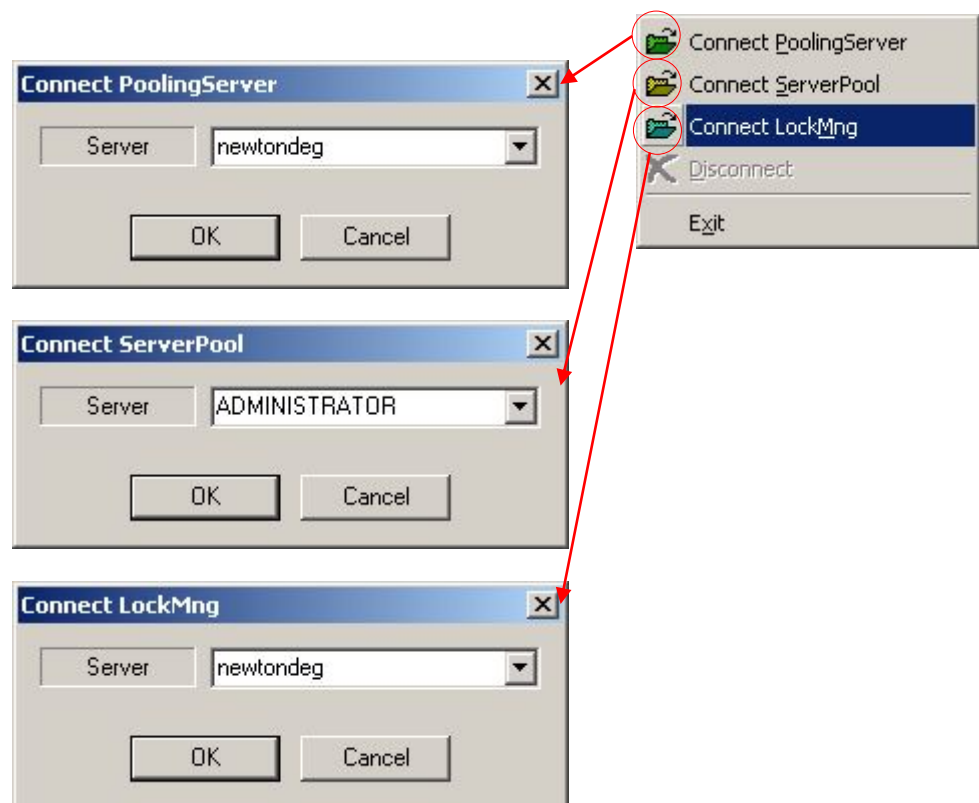


Abbildung 6: Dialogbaustein *Connect PoolingServer*, *Connect ServerPool* und *Connect LockMng*

Eingabe des Servernamens

Beim ersten Anmelden müssen Sie den Namen des Rechners angeben mit dem Sie sich verbinden wollen. Danach können Sie den Servername aus einer Liste auswählen.

- ☛ Klicken Sie danach auf OK.

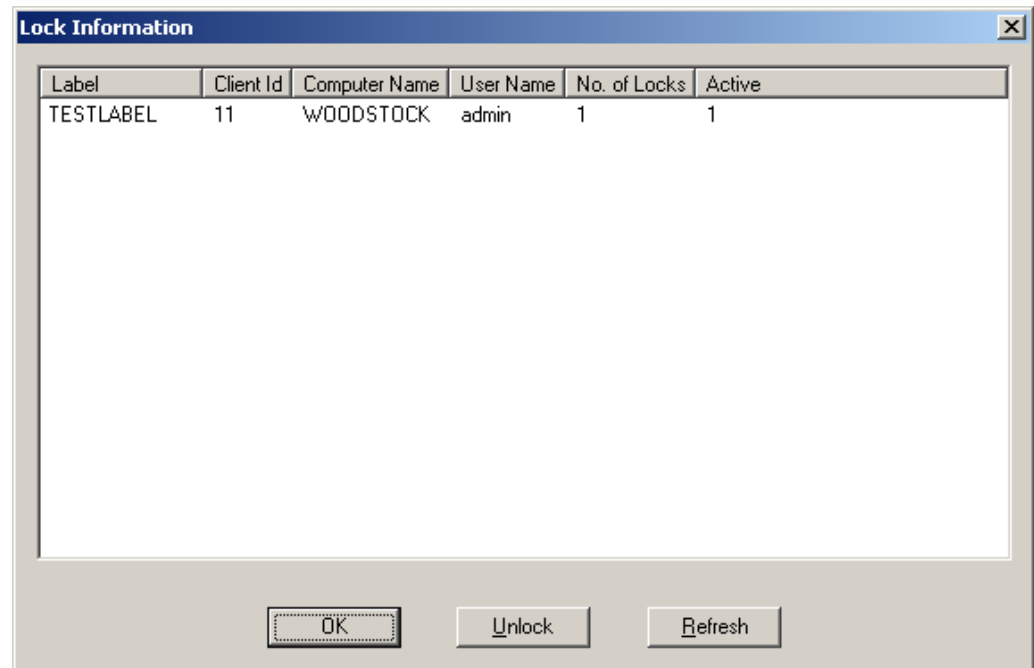


Abbildung 7: Dialogbaustein *Lock Information* in dem alle gelockten **V5** Transaktionen aufgeführt werden

In diesem Dialog werden die dauerhaften Datensperren angezeigt (gelockte Daten, werden im weiteren Verlauf als persistente *Locks* bezeichnet).

- Persistente Locks können nur nach einer gewissen Zeitspanne gelöscht werden. Aktive Transaktionen können nicht gelöscht werden. Deshalb ist ein Löschen erst dann möglich, wenn der Server merkt, dass ein Client nicht mehr verfügbar ist (Konfiguration). Danach können die Daten wieder freigegeben werden.
- Der Computernamen und der Benutzername werden nur bei persistenten Locks angezeigt.
- In der Spalte *No. Of Locks* wird die Anzahl der Objekte aufgelistet, die persistent gelockt sind.

Die Menüleiste und die Werkzeugleisten



Das **Connection Menü**

Über dieses Menü verbinden Sie sich mit einem Server, lösen die Verbindung oder verlassen die Server Tools.

Das **View Menü und das Windows Menü**

Über diese Menüs steuern Sie die Benutzeroberfläche der Server Tools.

Das **Process monitor Menü**

- ☐ **Show all Process:** Ist dieser Eintrag mit einem Häkchen versehen, werden alle Prozesse die auf dem Rechner laufen angezeigt. Sie werden hier die gleichen Prozesse wie im Task-Manager finden. Ist das Häkchen nicht gesetzt, werden nur die Manufacturing Hub-Serverprozesse angezeigt
- ☐ **Update speed:** Hier stellen Sie die Aktualisierungsgeschwindigkeit ein.

Das **Help Menü**

Über diese Menüs erhalten Sie die aktuelle Information über die installierte Version der Server Tools.

Arbeiten mit dem PoolingServer

Um alle Server anzuzeigen, gehen Sie so vor:



- In Ihrem Browser sehen Sie den Server auf dem Sie sich angemeldet haben. Öffnen Sie auf diesem Eintrag das Kontextmenü und wählen die Funktion **Show running ServerPools**. [Show running ServerPools](#)

Reload

- Mit einem Doppelklick auf den PoolingServer, oder mit einem nochmaligen Aufruf des Kontextmenüs und der Auswahl der Funktion **Reload**, werden Ihnen unterhalb des PoolingServer Knotens alle Server Pools aufgelistet.



- Öffnen Sie auf einem Eintrag das Kontextmenü und wählen die Funktion **Show Server Instances**. [Show Server Instances](#)

Reload

- Mit einem Doppelklick auf den Server, oder mit einem nochmaligen Aufruf des Kontextmenüs und der Auswahl der Funktion **Reload**, werden Ihnen alle Clientdaten dieser Server angezeigt.

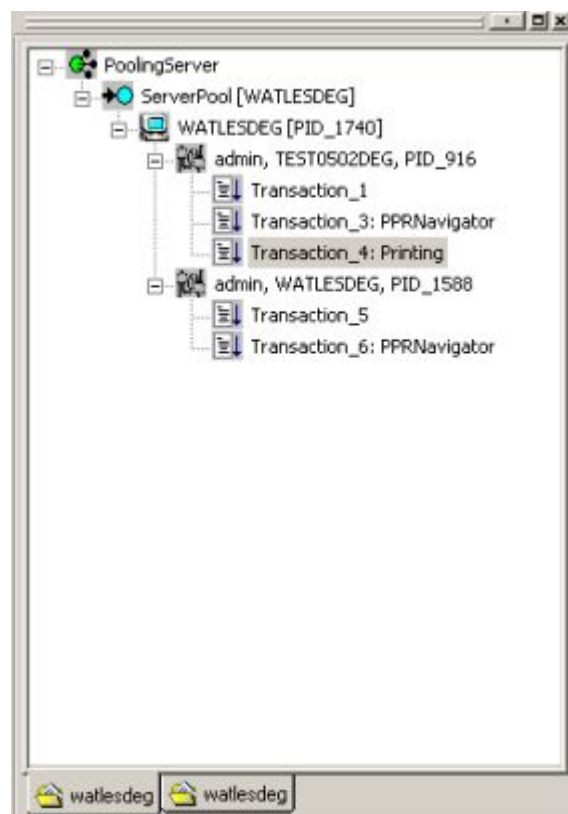


Abbildung 8: PoolingServer und Server Tools mit 2 angemeldeten Clients

Das Kontextmenü des PoolingServers

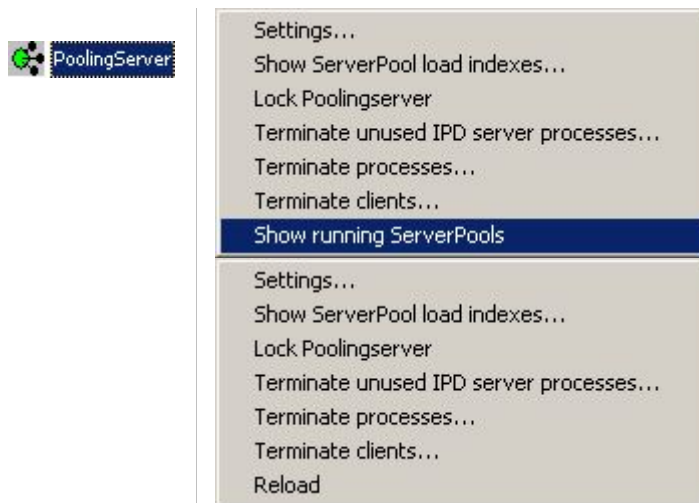


Abbildung 9: Kontextmenü des PoolingServer

Settings...

Das Kontextmenü Settings... (Einstellungen) des PoolingServers erlaubt das Editieren der PoolingServer Entscheidungskriterien, während des laufenden Betriebes. Wie Sie die Kriterien verändern können, haben Sie im Abschnitt [Konfigurierbare Lastkriterien](#) bereits erfahren.

Wenn Sie die Einstellungen mit „OK“ beenden, werden die neuen Werte für die gegenwärtigen Berechnungen verwendet.

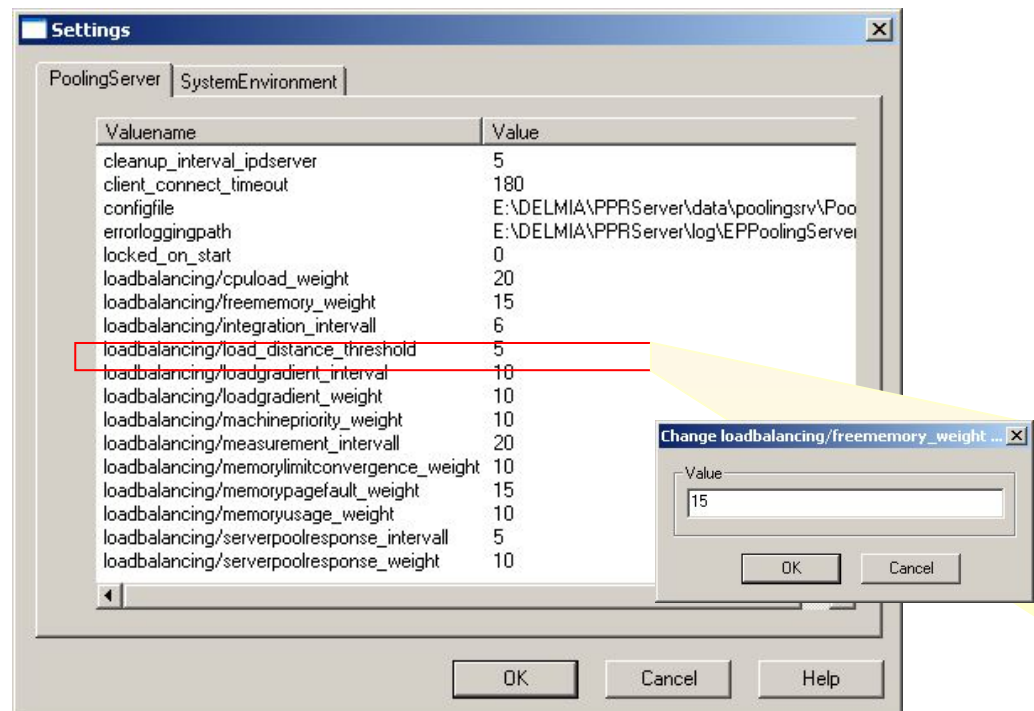


Abbildung 10: Eigenschaften des PoolingServers

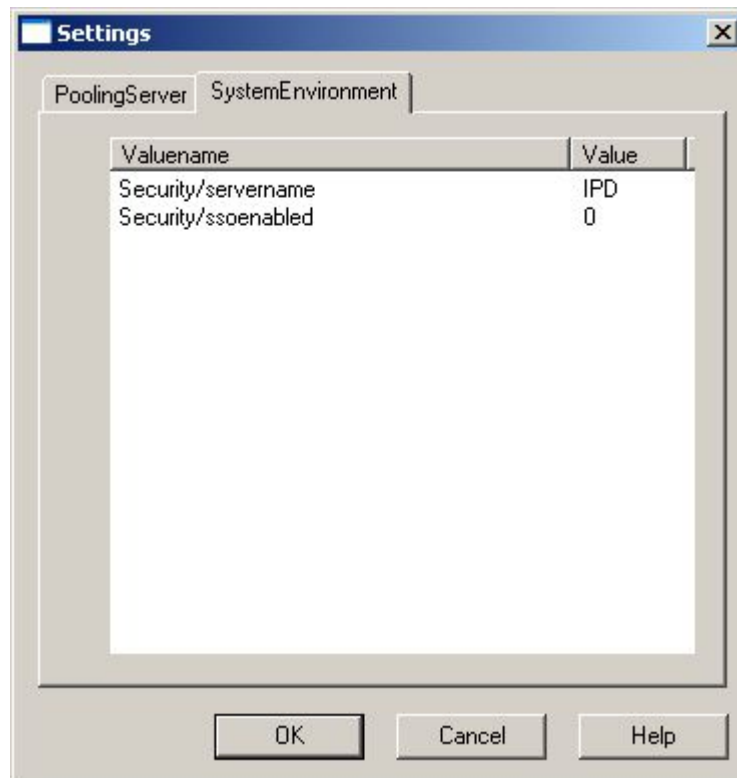


Abbildung 11: Eigenschaften des PoolingServers Administrations-Clients

Konfigurierbare Lastkriterien des PoolingServer

Es gibt mehrere Lastkriterien um die Zuteilung der Server an bestimmte Clients zu konfigurieren. Die Zuteilung der Server anhand der aktuellen Serverauslastung kann über die Gewichtung der Lastanteile konfiguriert werden. Durch Anpassung der Lastanteile für die Berechnung des Auslastungsindex der Servermaschine, können die tatsächlichen Rechnerausstattungen berücksichtigt werden.

Die „**weight**“ Kriterien stellen die relativen Anteile, mit denen die Messwerte gewichtet werden, dar. Sie sind über den Windows-Registrierungseditor änderbar - Registrierungsschlüssel:

In "HKEY_LOCAL_MACHINE\SOFTWARE\

"Delmia\Ergoplan PoolingServer\loadbalancing"

- **cpuload_weight [%]**
CPU Last der gesamten Maschine
- **freememory_weight [%]**
Verfügbarer freier virtueller Speicher des Rechners (negiert berücksichtigt)

- **Machinepriority_weight [%]**
Rechnerabhängige Priorität laut Poolingserver-Konfiguration (negiert berücksichtigt)
- **Loadgradient_weight [%]**
Gemessener Lastanstieg
- **Memorypagefault_weight [%]**
Speicherseitenfehler pro [sec]
- **Memorylimitconvergence_weight [%]**
Verhältnis des genutzten virtuellen Speichers des PPRServers zu dem konfigurierten Grenzwert des virtuellen Speicher der Server Tools.
- **Memoryusage_weight [%]**
Verhältnis des virtuellen Speichers des PPRServers zu dem frei verfügbaren virtuellen Speicher des Rechners.
- **Serverpoolresponse_weight [%]**
umgekehrtes Verhältnis der Serverpool-Antworten zu den erwarteten Antworten innerhalb des Messungsintervalls.

Um den Einfluss einer bestimmten Größe zu optimieren sollten sie höher gewichtet werden. Hat z. B. eine Servermaschine wenig verfügbaren virtuellen Speicher und eine niedrige CPU Last dann:

⇒ kann zur Optimierung der Wert von freememory_weight [%] höher gesetzt werden.

- **cleanup_interval_Manufacturing Hub-Server**
Mit 'cleanup_interval_Manufacturing Hub-Server' wird das Intervall für das Beenden von ungenutzten Manufacturing Hub-Server-Prozessen in Minuten festgelegt.
'0' schaltet die Funktion ab.

Mit den folgenden Einstellungen kann das Intervall der Messzeit modifiziert werden:

- **serverpoolresponse_intervall**
Antwortintervall der Serverpools in Sekunden
- **measurement_intervall [sec]**
Messintervall für die Leistungsdaten der Server Pools
- **integration_intervall [measure-cycles]**
PoolingServer – Integrationsintervall für die Messdaten in Messzyklen
- **loadgradient_intervall [integ-cycles]**
Integrationsintervall des Lastanstiegs

Umso kleiner die Intervalle der Messzeit sind, um so mehr werden kurzfristige Schwankungen der Lastdaten des ServerPools berücksichtigt. Umgekehrt gilt: je größer die Messintervalle, um so mehr werden die langfristigen Lastwerte in der Berechnung des Lastindex berücksichtigt – kurzfristige Schwankungen spiegeln sich dagegen weniger wieder.

- **locked_on_start**

Wenn der Registrierungseditor-Eintrag *locked_on_start* auf '1' gesetzt ist, startet der Poolingserver im gesperrten Modus, d. h. es können sich keine Clients (ausgenommen Administratoren-Clients), auf den PPR-Server verbinden.

Schlüssel: HKEY_LOCAL_MACHINE\SOFTWARE\Delmia\Ergoplan PoolingServer
Eintrag: locked_on_start
Wert: 0

Standardmäßig wird der Wert 0 gesetzt.

Der Wert kann auch durch einen Batch-Aufruf des ServerTools-Clients gesetzt werden:

```
- locked_on_start = 1: dpfframe.exe /dpfres:epservertoolsres.dll  
/poolingserver_lock_on_start  
- locked_on_start = 0: dpfframe.exe /dpfres:epservertoolsres.dll  
/poolingserver_unlock_on_start
```

Analog kann ein laufender PoolingServer in den Lockzustand versetzt werden:

```
- locked: dpfframe.exe /dpfres:epservertoolsres.dll  
/poolingserver_lock  
- unlocked: dpfframe.exe /dpfres:epservertoolsres.dll  
/poolingserver_unlock
```

▪ load_distance_threshold

Schlüssel: HKEY_LOCAL_MACHINE\SOFTWARE \ DELMIA \ ERGOPlan PoolingServer \ loadbalancing
Eintrag: load_distance_threshold
Wert: 5

Der Eintrag 'load_distance_threshold' legt den Schwellwert der Lastdifferenz verfügbarer Servermaschinen fest, ab dem das Loadbalancing auf Basis von Maschinenlastmessungen angewendet wird. Liegt die Lastdifferenz unter diesem Schwellwert, werden die verfügbaren Servermaschinen auf Basis der Anzahl der verbundenen Clients gesteuert. Dabei wird die Maschine des für die Clientverbindung konfigurierten Server-Subnetztes verwendet, die die kleinste Anzahl an Clientverbindungen hat. Die Lastdifferenz wird aus dem maximalen und minimalen ermittelten Lastindex der für einen Clientanmeldung konfigurierten und verfügbaren Maschinen berechnet. Eine Maschine ist für das Lastbalancing verfügbar, wenn sie kontinuierlich Lastmesswerte an den Pooling-Server liefern.

Anwendung des Registrierungseintrages

Sind gleichzeitig viele Clients mit dem Server verbunden, ist die Anzahl der Clientverbindungen ein potentieller Last- und Ressourcenfaktor, auch wenn unmittelbar nach dem Verbinden noch keine Last auf der Servermaschine erzeugt wird.

Auswirkungen auf das PoolingServer-Lastbalancingverfahren:

- Mit dem Wert '0' verhält sich das Lastbalancing wie vor DPE5.16.
- Der Standardwert '5' bewirkt, dass in DPE5.16 erst ab einer Lastindexdifferenz von 5 (im ServerTools-Client angezeigte Lastindexwerte für Servermaschinen) der Client zu der Maschine mit dem kleinsten Lastindex verbunden wird. Ist die Lastdifferenz kleiner 5, wird der Clientverbindung anhand der Anzahl der Clientverbindungen der Servermaschinen balanciert. Es wird die Maschine verwendet, die die kleinste Anzahl verbundener Clients aufweist.
- Wird der Wert auf '100' (Summe der in der Registry konfigurierten Lastanteile) oder größer gesetzt, wird das Lastbalancing ausschließlich auf Basis der Anzahl der verbundenen Clients durchgeführt.

Anpassung des Schwellwertes 'load_distance_threshold':

Der Schwellwert wird in Installationen mit einer hohen Anzahl von inaktiven Clientverbindungen benötigt, um die Verteilung der Clients über alle Maschinen optimieren zu können. Ein Client ist für das Lastbalancing inaktiv, wenn nach Verbinden zu einem Manufacturing Hub-Server die Servermaschine nicht messbar belastet wird. Ist das bei einer hohen Anzahl von Clients der Fall, wird sich der Lastindex zum Verbindungszeitpunkt nur unwesentlich unterscheiden. Die Verteilung der Clients wäre für diesen Fall von kurzfristigen kleinen Schwankungen des Lastindex bestimmt und kann nach Beginn lastintensiver Aktivitäten zu einer Fehlverteilung der Last führen.

Der Defaultwert '5' ist ein Minimalwert (bezogen auf einen maximalen Lastindexwert von 100), der das Lastbalancing gegen eine Fehlverteilung von Clientverbindungen bei geringen Lastunterschieden absichert. Mit einem höheren Wert wird die Anzahl der Clientverbindungen stärker gegenüber der tatsächlichen Maschinenlast zum Verbindungszeitpunkt (CPU, Memory) berücksichtigt. Das ist für Installationen sinnvoll, die Lastspitzen erst deutlich nach Anmelden der meisten Clients erreichen. Um jedoch eine Überlastung einzelner Maschinen bereits während des Verbindens neuer Clients zu vermeiden, sollte der Wert 30% des maximalen Lastindex nicht überschreiten. Der max. Lastindex ist die Summe der in dem Registrierungseditor konfigurierten Lastanteile (weightings).

Wann sollten die Kriterien geändert werden?

Die Standardeinstellungen sind unkritisch und ausreichend für Lastmessungen normal ausgelasteter Server. Anpassungen sind nur erforderlich, wenn sich Unterschiede im tatsächlichen Lastverhalten der Server relativ zueinander nicht in unterschiedlichen Lastindexen widerspiegeln. Ursache dafür kann eine gleichmäßig überproportional starke Belastung aller Server für ein einzelnes Kriterium sein. Bei Änderungen der Lastgewichtungen müssen die tatsächliche Ausstattung der Rechner und die beobachteten Auslastungswerte in den Einzellastkriterien (Erfahrungswerte des Administrators) berücksichtigt werden.

Beispiel**Was sind nun spezielle Rechnerausstattungen?**

Haben alle Server einen gleich geringen virtueller Speicher, kann durch das Herabsetzen des **freememory_weight** Kriteriums die PoolingServer Entscheidung optimiert werden. Durch die geringere Gewichtung werden andere Kriterien für die Entscheidung wichtiger.

Gleiches kann bei gleichmäßig hohen Speicherseitenfehlern (**Memorypage-fault_weight**) aller Server gesagt werden. Werden bei geringer Speicherausstattung beständig RAM-Speicherseiten in *Auslagerungsdateien geschrieben*, entsteht eine hohe Speicherseitenfehlerrate. Sind davon alle Servermaschinen betroffen, kann eine geringere Gewichtung die PoolingServer Entscheidung verbessern. Hohe Speicherseitenfehler treten jedoch im Normalfall nur kurzfristig auf, so dass der oben genannte Fall eher auf eine zu geringe RAM-Speicherausstattung schließen lässt.

Allgemein kann davon ausgegangen werden, dass das Measurement_intervall [sec] ein Vielfaches (≥ 4) des Serverpoolresponse_intervalls [sec] sein sollte. Der PoolingServer zählt die Antworten der Server Tools, um die allgemeine Zuverlässigkeit des Antwortzeitverhaltens der PPRServer im Netz zu beurteilen.

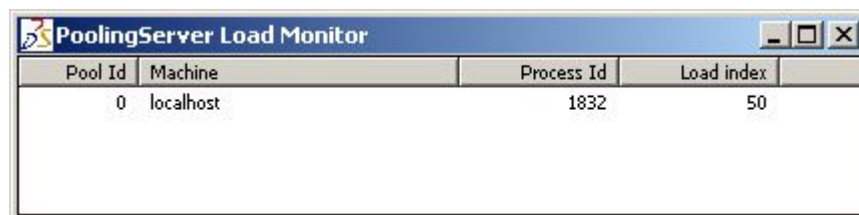
Auswirkungen der Veränderung von Messintervallen:

- ⇒ Umso höher das Integration_intervall [measure-cycles] eingestellt ist, umso mehr wird die Langzeit Maschinenlast bei die Entscheidung berücksichtigt.
- ⇒ Je höher das Loadgradient_interval [Integ-cycles] gewählt wurde, umso mehr wird das Langzeit-Lastgefälle berücksichtigt.
- Die Summe aller Kriterien muss 100% sein.
- Der Lastindex ist ein relativer Wert mit einem Bereich von 0 bis 100.

Ein höherer Lastindex zeigt eine höhere Maschinenlast für einen PPR Server an, im Vergleich mit den weiteren aufgelisteten Servermaschinen.

Show ServerPool load indexes...

Durch Wählen von „Show ServerPool load indexes...“ öffnet sich ein Dialog, in dem alle von den ServerPools berechneten Lastdaten der gegenwärtigen Lastindices angezeigt werden.



Pool Id	Machine	Process Id	Load index
0	localhost	1832	50

Abbildung 12: „Show ServerPool load indexes“ des Pooling Servers

Lock PoolingServer

Wenn Sie „Lock PoolingServer“ wählen, wird die Neuanmeldung eines Clients unterbunden. Es können sich keine Clients mehr anmelden.

Mit **Unlock Poolingserver**, im gleichen Kontextmenü, können Sie die Anmeldung wieder gestatten.

Terminate unused Manufacturing Hub server processes...

Mit der Funktion „Terminate unused IPD server processes“ werden diejenigen Manufacturing Hub-Server-Prozesse beendet, zu denen keine PE-Clients mehr verbunden sind.

Die Clientverbindung zum Server wird entweder beim regulären Beenden des Clients oder beim Beenden des Clients über den ServerTools-Client (Kontextmenü *Terminate Client*) getrennt. Siehe auch [Beenden von Clients](#).



Hinweis:

Damit der PE-Client nach Aufruf von „Terminate Client“ in den ServerTool vom Server getrennt werden kann, muss die Meldung „Verbindung zum Server wurde getrennt“ im PE-Client geschlossen werden, ansonsten ist die Trennung der Verbindung für das Terminieren ungenutzter Manufacturing Hub-Server-Prozesse unwirksam.

Wurde der Client über das Kontextmenü „Terminate Client“ vom Server getrennt, wird nach einer Zeitspanne von 6-8 min. die Clientverbindung im ServerPool ungültig, d.h. erst nach dieser Zeitspanne kann der betreffende Manufacturing Hub-Server-Prozess mit der Funktion „nicht verwendete Manufacturing Hub-Server terminieren“ beendet werden.

Die Funktion „Terminate unused IPD server processes“ kann über das Kontextmenü oder in einem Batchdatei aufgerufen werden.

Der Aufruf in der Batchdatei lautet:

```
DPFFrame.exe /dpfres:epserverToolsres.dll /gc_Manufacturing  
Hub-Server
```

Im Registrierungseditor (- für Batch -Terminierung abgefragt):

Schlüssel

HKEY_LOCAL_MACHINE\SOFTWARE\Delmia\EPSToolsUI

Eintrag: PoolingServer

Wert: <machine>

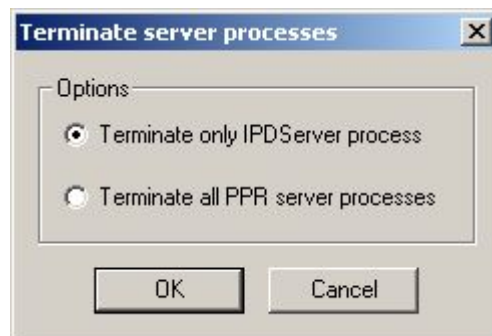
<machine> ist der Name des Computers, auf dem der PoolingServer läuft.

Terminate processes...

Wenn Sie „Terminate processes“ wählen, erscheint ein Dialog in dem Sie festlegen können, ob alle PPR Serverprozesse oder der IPD Serverprozesse beendet werden sollten.

⇒ Beide Fälle werden auf allen Server Rechnern ausgeführt.

Beim Wählen „all PPR server processes“ werden der PPR Update - , Lock - und Rights Manager auf dem Master PPR Server Rechner beendet. Die Server Tools und der PoolingServer werden nicht beendet, da sie für die Server Tools Benutzeroberfläche benötigt werden.



Zusätzlich werden alle EPGenericServices.exe-Prozesse beendet.

Daraus leiten sich neue Batch-Commands benötigt:

a) `/kill_all_pprserver_processes`

⇒ terminiert alle PPRServer-Prozesse (EPGenericServices.exe, Manufacturing Hub-Server.exe, LockMng.exe, UpdateMng.exe) auf allen Maschinen

b) `/kill_all_Manufacturing Hub-Server_processes`

⇒ terminiert alle Manufacturing Hub-Server-Prozesse (EPGenericServices.exe, Manufacturing Hub-Server.exe) auf allen Maschinen

Beispiel:

```
\bin\DPFFrame.exe /dpfres:epservertoolsres.dll /kill_all_Manufacturing Hub-Server_processes
```

Terminate clients...

Hiermit beenden Sie alle Clientprozesse der DPE Installation. Siehe auch [Beenden von Clients](#).

Reload oder Show running ServerPools

Die PoolingServer-Daten werden nicht automatisch aktualisiert. Soll der aktuelle Stand angezeigt werden, führen Sie Reload aus.

Abschalten der Lastmessung von PPR-Servermaschinen

Die Lastmessung auf Basis von Windows-Performance-Indikatoren kann für eine PPR-Serverdomäne (Gruppe von PPR-Servermaschinen) in der PoolingServer-Konfiguration abgeschaltet werden. Die Änderung der Konfiguration ist während der Laufzeit des PoolingServer möglich. Die Änderungen sind nach max. 30 sec. wirksam.

Die Performance-Lastmessung sollte für PPR-Servermaschinen abgeschaltet werden, bei denen Windows keine Performance-Indikatoren für Prozesse unterstützt bzw. die System-Performance-DLL abgeschaltet ist. Die Verfügbarkeit der Windows-Performance-DLL kann im Registrierungseditor abgefragt werden:

Schlüssel

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfProc\Performance

Eintrag: Disable Performance Counters

Wert: 0 - verfügbar, ansonsten nicht verfügbar

Wenn der Wert in dem Registrierungseditor nicht 0 ist, muss für diesen Rechner [disable_load_measurement](#) in der Poolingserver-Konfigurationsdatei gesetzt werden. Erst jetzt können sich PPR-Clients über den PoolingServer auf einen PPRServer verbinden.

Beispiel aus der Poolingserver-Konfigurationsdatei:

Beispiel

```
<netconfig>>
// - optional parameter: @opt
// - <option>: disable_load_measurement - disables the performance data based load measurement
servernet:
    Location [localhost(0)] @opt = { disable_load_measurement }
clientnet:
```

Für Clients, bei denen die Lastmessung abgeschaltet ist, werden keine Prozessinformationen angezeigt (ServerTools). Die Lastindexe dieser Maschinen werden aus den Indikatoren:

- ⇒ Anzahl der zu dieser Maschine verbundenen PPR-Clients - 40%
Gewichtung
- ⇒ Netzwerk-Antwortzeit des ServerPools - 30%
Gewichtung
- ⇒ Maschinenpriorität (PoolingServer-Konfiguration) - 20%
Gewichtung
- ⇒ Lastanstieg (aus den anderen Indikatoren berechnet) -10%
Gewichtung

berechnet und im ServerTools-Client in [] -Klammern gesetzt angezeigt. Es können nur Lastindexe verglichen werden, denen dieselbe Berechnungsmethode zugrunde liegt, d. h. nur Indexe ohne [] oder nur Indexe mit [].

Was geschieht mit Clients die nicht in der Konfigurationsdatei aufgelistet sind?

Nicht aufgelisteten Clients

Clients, die nicht im Abschnitt 'clientnet' genannt werden, wurden bis zur Version PE 5.14 immer potentiell auf alle Servermaschinen verbunden.

Ab Version PE 5.14 kann in der PoolingServer-Konfiguration eine Gruppe von Servern konfiguriert werden, zu denen sich alle **nicht aufgelisteten** Clients verbinden.

```
PoolingServer-Konfigurationseintrag:
...
servernet:
    @anonymous_userdomain [server1, server2]
...
```

Der Eintrag ist u. a. dann nützlich, wenn man zwischen mehreren Nutzergruppen (Administratoren, Bearbeiter) unterscheidet, aber nicht alle Clientmaschinen einer dieser Gruppen (z. B. Bearbeiter) eintragen will oder zu häufig ändern müsste.

Das Kontextmenü des ServerPools

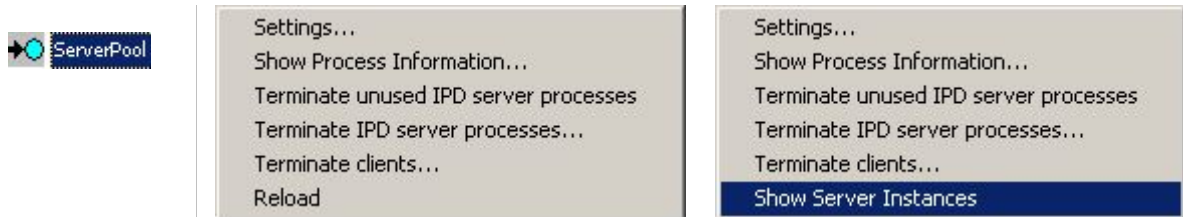


Abbildung 13: Kontextmenü des ServerPools

Serverpool-Einstellungen (Settings...)

Über diese Einstellungen können Sie die Registrierungseinträge des Server-Rechners bearbeiten.

Der Serverpool verwaltet die Manufacturing Hub-Serverprozesse, wobei bestimmte Schwellwerte für die Zuteilung eines neuen Serverprozesses an den Client eingehalten werden. So wird bei Überschreitung der Grenzwerte für die Anzahl der Clients und der Auslastung des virtuellen Speichers ein neuer Server gestartet.

Ein Serverprozess kann nur bis zur Betriebssystemgrenze von 2 GB virtuellen Speicher arbeiten. Bis zu dieser Grenze kann das Betriebssystem Speicher auslagern. Wenn diese Grenze überschritten wird, kann ein Serverprozess nicht weiterbearbeitet werden. Der Prozess kommt zum Stillstand. Die Einstellungen, die Sie unter dem Reiter *Server Tools* vornehmen können, betreffen den Serverpool.

Werden die Grenzwerte für die Anzahl der Clients (serverpool/clientlimit) und den virtuellen Speicher (serverpool/memorylimit) über den ServerTools - Eingabedialog geändert, sind die Änderungen auch für einen bereits gestarteten Serverpool wirksam.

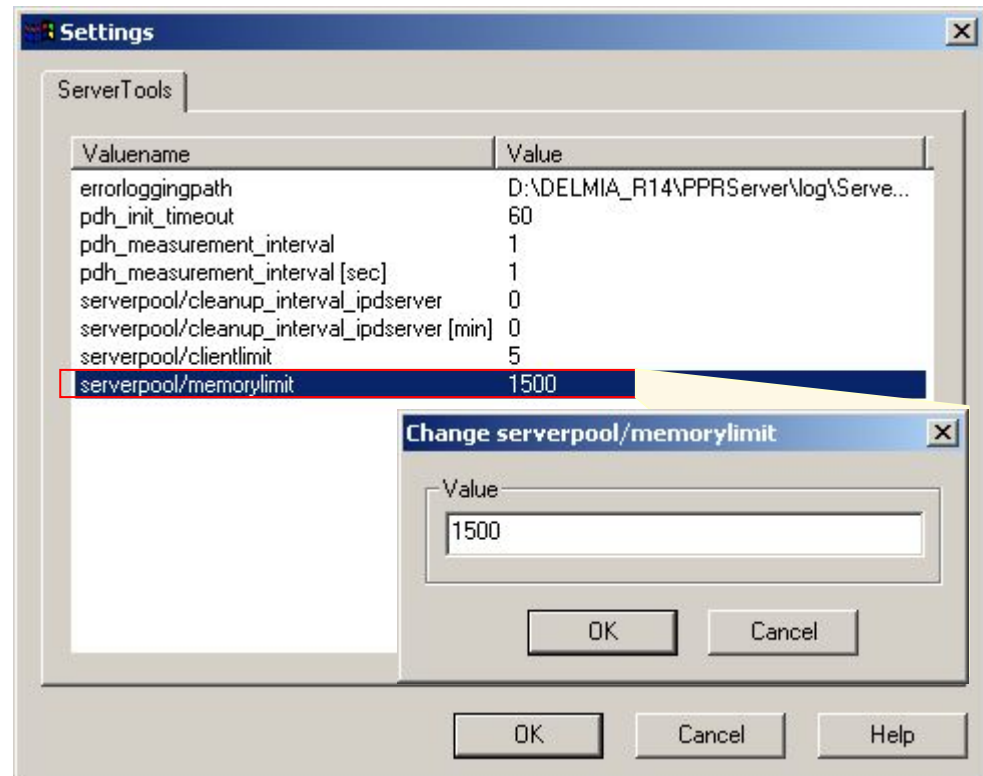


Abbildung 14: Eigenschaften des ServerPools

☐ **pdh_measurement_interval [sec]**

Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\Delmia\Ergoplan EPServerTools

Eintrag: pdh_measurement_interval [sec]

Wert: '1'

Der ServerPool Registrierungs-Eintrag *pdh_measurement_interval* legt die Zeitspanne fest, mit dem Performance-Daten der Maschine und von den PPRServer-Prozessen abgefragt werden. Das Minimum ist 1 sec. Man kann den Wert erhöhen, um die CPU-Last der EPServerTools.exe zu verringern, wenn viele Prozessmonitore der ServerTools-Clients parallel laufen und so eine hohe CPU-Last generieren.

Der Eintrag *pdh_measurement_interval* beeinflusst die Aktualisierung der Prozessinformationen im ServerTools-Client und der Maschinen-Lastberechnung des PoolingServers. Wird ein zu hoher Wert verwendet (z. B. 20 sec.) kann der Lastindex durch den PoolingServer nicht ausreichend genau (abhängig von LoadBalancing-Konfiguration) ermittelt werden.

⇒ Normaler Einstellbereich ist: 1...5 sec.

☐ **Manufacturing Hub-Server_availability_check_timeout**

Schlüssel HKEY_LOCAL_MACHINE \ SOFTWARE \ Delmia \ Ergoplan EPServerTools \ serverpool

Eintrag: Manufacturing Hub-Server_availability_check_timeout

Wert: '30'

Der Wert gibt den Timeout in [Sekunden] für den Test der Manufacturing Hub-Server - Verfügbarkeit an. Bei diesem Test wird versucht, einen COM RPC-Aufruf auf den zu testenden Manufacturing Hub-Server-Prozess auszuführen.

- ⇒ Ist dieser Test vor Ablauf des Timeouts nicht erfolgreich, gilt der Manufacturing Hub-Server als nicht verfügbar und wird im ServerPool für Client-connects nicht mehr verwendet.
Der Timeout begrenzt somit die Zeitspanne für einen solchen Test, der bei einer Clientanmeldung für den nächsten, zu verbindenden Manufacturing Hub-Serverprozess durchgeführt wird. Die maximale Wartezeit eines Clientconnects zu einem ServerPool kann dann mit folgender Formel berechnet werden:

$$t_{max_Connect}[Sek] =$$

$$\text{Anzahl_der_Manufacturing Hub-Serverprozesse} \times \text{Manufacturing Hub-Server_availability_check_timeout} + \text{Startzeit_eines_Manufacturing Hub-Serverprozesses [Sek]}$$

In der Regel wird die maximale Wartezeit auch bei mehreren zum Verbindungszeitpunkt hoch ausgelasteten Serverprozessen nicht erreicht, da ab DPE5.16 eine Servicerroutine des ServerPools periodisch im Abstand von 10 Sekunden prüft, ob Manufacturing Hub-Serverprozesse nicht mehr verfügbar sind und diese Prozesse für die nächsten Clientverbindungen nicht mehr berücksichtigt werden. Die durchschnittliche Client-Connectzeit wird somit zum überwiegenden Teil durch den Zeitbedarf für das Starten neuer Serverprozesse bestimmt und ist für den Fall der Wiederverwendung eines Manufacturing Hub-Serverprozesses vernachlässigbar.

❑ Manufacturing Hub-Server_creation_timeout

Schlüssel HKEY_LOCAL_MACHINE \ SOFTWARE \ Delmia \ Ergoplan EPSTools \ serverpool

Eintrag: Manufacturing Hub-Server_creation_timeout

Wert: '120'

Der Serverpool-Eintrag 'Manufacturing Hub-Server_creation_timeout' gibt den Timeout in [Sekunden] für das Erzeugen eines neuen Manufacturing Hub-Server-Prozesses im ServerPool an. Wenn bei einer Clientverbindung innerhalb dieser Zeit keine neue Manufacturing Hub-Server-Instanz (== COM-Komponente im neuen Manufacturing Hub-Server-Prozess) verbunden werden konnte, gibt der ServerPool eine Fehlermeldung zurück, dass dieser Timeout überschritten wurde. Verbindet der Serverpool einen bereits bestehenden Manufacturing Hub-Server-Prozess, wird kein Timeout angewendet. Der Timeout für das Erzeugen neuer Manufacturing Hub-Server-Prozesse verhindert Connect-Blockaden, wenn eine aktuell überlastete Maschine zu lange zum Initialisieren eines Manufacturing Hub-Servers braucht oder das Initialisieren des Manufacturing Hub-Servers durch eine wartende Datenbankverbindung (DB-Fehler oder Ressourcenmangel) blockiert ist.

❑ client_connect_timeout

Schlüssel HKEY_LOCAL_MACHINE \ SOFTWARE \ Delmia \ Ergoplan EPSTools \ serverpool

Eintrag: client_connect_timeout

Wert: '180'

Der PoolingServer-Eintrag 'client_connect_timeout' gibt den Timeout in [Sekunden] für das Verbinden eines Clients mit einem Manufacturing Hub-Server über einen ServerPool an. Kann ein Client innerhalb dieser Zeit nicht mit einem Manufacturing Hub-Server auf der vom Lastbalancing ausgewählten Maschine verbunden werden, wird eine Timeout Fehlermeldung an den Client zurückgegeben. Der Client kann dann einen zweiten Verbindungsversuch starten. Mit dem Timeout wird verhindert, dass nachfolgende, wartende Client-connects blockiert werden, wenn eine ServerPool-Maschine die Verbindungsanforderung nicht zurückgibt (wartet oder "hängt").

Serverprozess Lastwerte

❑ serverpool/clientlimit

Über den Eintrag *serverpool/clientlimit* legen Sie fest, ab welcher Anzahl von Clients ein neuer Server gestartet wird. Für den neuen Server gelten dann auch die 2 GB Betriebssystemgrenze.

❑ serverpool/memorylimit

Über den Eintrag *serverpool/memorylimit* legen Sie fest, ab welcher Auslastung des virtuellen Speichers ein neuer Server gestartet wird.

An einigen Beispielen soll der Serverpool erklärt werden.

- Im **Beispiel 1** steht Ihnen eine Maschine mit 4 GB (4.000 MB) Speicher zur Verfügung.
- Die Standardeinstellung für *serverpool/clientlimit* = 5, und wird auch so belassen.
- Das Tagesarbeitsszenario sieht so aus, dass 10 Clients sich bei Arbeitsbeginn bei dem Server anmelden.

Jeder einzelne Client verbraucht im Laufe des Tages Prozessspeicher des Servers, da seine Datenobjekte vom Server verwaltet werden. Mit der Standardeinstellung von 5 werden 2 Server gestartet sein und der gesamtnutzbare Prozessspeicher beträgt 4 GB. Der Erfahrungswert des Administrators, dass 4 GB virtueller Prozessspeicherverbrauchs ausreichen, wird hiermit bestätigt.

Meldet sich nun ein weiterer Client am Serverpool an, wird ihm ein neuer Serverprozess mit weiteren 2 GB zugeteilt. Somit ist gewährleistet dass ausreichend Prozessspeicher zur Verfügung gestellt wird. Wenn der genutzte Prozessspeicher den physikalischen RAM-Speicher überschreitet, übernimmt das Betriebssystem die notwendige Speicherauslagerung.

In einem weiteren Beispiel soll die *memorylimit*-Grenze beschrieben werden.

- Im **Beispiel 2** steht Ihnen eine Maschine mit 1 GB (1.000 MB) Speicher zur Verfügung.

Die Standardeinstellung für *serverpool/memorylimit*= 1500, und wird auch so belassen.

Falls zu wenig physikalischer RAM-Speicher zur Verfügung steht, muss das Betriebssystem Speicher auslagern. Dadurch wird das Antwortzeitverhalten des Manufacturing Hub-Servers beeinträchtigt. Dies kann verbessert werden, indem man ein *memorylimit* verwendet, das nur unwesentlich über dem im Tagesdurchschnitt zur Verfügung stehenden Real-Speichers liegt. Den verfügbaren Realspeicher können Sie im Task-Manager einsehen.

Realer Speicher (KB)	
Insgesamt	1048032
Verfügbar	592316
Systemcache	363656

Als eine Richtgröße kann der verfügbare Real-Speicher + 50% verwendet werden. In dem Beispiel ist, als erster Anhaltswert, der Standardwert also richtig. Wenn im Beispiel 4 Clients bereits 1500 MB Prozessspeicher verbrauchen, wird dem fünften Client ein neuer Serverprozess zugewiesen.

Für den Fall das *ein* Client den gesamten Prozessspeicher verbraucht, gewährleistet der Grenzwert **memorylimit** von 1500 MB, dass der Server bei einer Clientanzahl unter dem Clientlimit betriebsfähig bleibt.

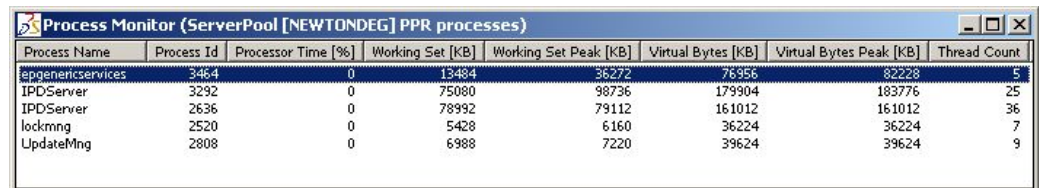
Der Grenzwert **memorylimit** sollte jedoch immer deutlich unter der Betriebssystemgrenze von 2 GB liegen und nicht zu klein gewählt werden. Ein zu kleiner Grenzwert führt zum Starten zu vieler Serverprozesse, die im Gesamtszenario wiederum mehr Speicher als ein einzelner Serverprozess verbrauchen, da jeder Serverprozess Verwaltungsdaten benötigt und Datencaches anlegt. Die maximal zu erwartende Anzahl von Serverprozessen sollte nicht größer als 4 je 2GB RAM sein. Die genaue Anzahl lässt sich aus der Speicherbelegung des Manufacturing Hub-Serverprozesses unmittelbar nach dem Start ermitteln. Der summierte Startverbrauch an Speicherseiten (nicht virtuelle Größe), über die maximal zu erwartende Anzahl von Serverprozessen, sollte nicht größer als ca. 30% - 40% des frei verfügbaren RAM-Speichers der Maschine bei Systemstart sein.

Wenn Sie die Serverpools Einstelllungen über den Button OK verlassen, werden alle Eingaben gespeichert und auch für alle laufenden Prozesse verwendet.

Show Process Information...

Über diesen Menüpunkt werden die Prozessleistungsdaten angezeigt.

Wichtig ist die Spalte *virtuelle Bytes [KB]*. Sie zeigt den von einem PPR Prozess verwendete virtuelle Speicher. Die virtuelle Speichergrenze wird von Windows auf 2 GB beschränkt. Wenn diese Größe erreicht wird, kann ein PPR Prozess nicht mehr funktionieren. Um dies zu verhindern müssen Sie, wie im vorhergehenden Abschnitt beschrieben, die Speichergrenze anpassen.



Process Name	Process Id	Processor Time [%]	Working Set [KB]	Working Set Peak [KB]	Virtual Bytes [KB]	Virtual Bytes Peak [KB]	Thread Count
epgenericServices	3484	0	13484	36272	76956	82228	5
IPDServer	3292	0	75080	98736	179904	183776	25
IPDServer	2636	0	78992	79112	161012	161012	36
lockmg	2520	0	5428	6160	36224	36224	7
UpdateMng	2808	0	6988	7220	39624	39624	9

Abbildung 15: Show Process Information

Ab der Version PE 5.15 wird zusätzlich der PPR-Prozesses "EPGenericServices.exe" angezeigt.

Terminate unused IPD server processes...

Terminieren nicht verwendeter Manufacturing Hub-Server-Prozesse

War es in den bisherigen Versionen so, dass beim Aufruf der Windows-Funktion "KillProcess" alle nicht verwendeten Manufacturing Hub-Server-Prozesse sofort terminiert wurden (alle Aufräumarbeiten wurden von Windows durchgeführt), hat sich dieses Verhalten mit der neuen Version geändert..

Der ServerPool versucht zunächst einen Shutdown nicht verwendeter Manufacturing Hub-Server durchzuführen, wobei jeder Manufacturing Hub-Server-Prozess seine COM-Services und dann sich selbst beendet.

- ⇒ Falls der Shutdown eines Manufacturing Hub-Server-Prozesses innerhalb von 15 sec. nicht erfolgreich ist, wird dieser Prozess wie bisher terminiert.

Der dazugehörige Batch-Aufruf für „Terminieren nichtverwendeter Manufacturing Hub-Server-Prozesse“ lautet: `/gc_Manufacturing Hub-Server`

Siehe auch [Beenden von Clients](#) und nachfolgenden Kontextmenüeintrag.

Terminate IPD server processes...

Bevor Sie alle Server Prozesse (alle Manufacturing Hub-Server.exe) beenden, sollten Sie alle Clientprozesse dieses Servers (des selektierten ServerPools) beenden. Siehe auch [Beenden von Clients](#) und nachfolgenden Kontextmenüeintrag.



Zusätzlich werden auch alle PPR-Prozesse "EPGenericServices.exe" terminiert.

Terminate clients...

Hiermit beenden Sie alle Clientprozesse dieses Servers. Siehe auch [Beenden von Clients](#).

Reload

Die ServerPools-Daten werden nicht automatisch aktualisiert. Soll der aktuelle Stand angezeigt werden, führen Sie Reload aus.

Die Server Tools

Die ServerTools sind eine Administrationskonsole zur Überwachung und Konfiguration der Multiserver-Installation.

Mit den **Server Tools** können Sie die Manufacturing Hub-Serverprozesse überwachen oder Informationen zu den verbundenen Clients erhalten. IPD-Prozesse sind die Prozesse die auf der Integrierten Prozess Datenbank des DELMIA Process Engineers ablaufen. Mit den Server Tools können Sie, auch von einem anderen Rechner, die Serverprozesse überwachen.

Der Serverpool ist Teil des ServerProzesses „EPSTools.exe“ und stellt ein Serverprozess-Pooling für die IPD-Clients bereit. Für Manufacturing Hub-Serverprozesse werden Lastgrenzwerte definiert. Das Pooling-Verfahren gewährleistet, dass einem neu angemeldeten Client ein Manufacturing Hub-Serverprozess zugewiesen wird, der die Lastgrenzwerte noch nicht überschritten hat.

Wenn Sie den Task-Manager aus der Windows NT oder 2000 Umgebung kennen, werden Sie sich mit den Server Tools sehr schnell zurechtfinden.

Der ServerTools Prozess dient als Server für den ServerTools-Client und stellt den ServerPool für Manufacturing Hub-Server-Prozesse bereit. Nachfolgende Tabelle zeigt eine Übersicht der grundlegenden Funktionalität des ServerProzesses „EPSTools.exe“:

Server Tools Prozess besteht aus	
Server Tools-Server	Server Pool
<ul style="list-style-type: none"> ▪ Ermittelt und speichert Leistungsdaten von Prozessen und Maschine in einem Cache ▪ Liefert Prozess-Leistungsdaten an den ServerTools-Client 	<ul style="list-style-type: none"> ▪ Überträgt Maschinen- und Prozess-Leistungsdaten an den PoolingServer ▪ Weist dem PPR-Client ein Server Prozess zu

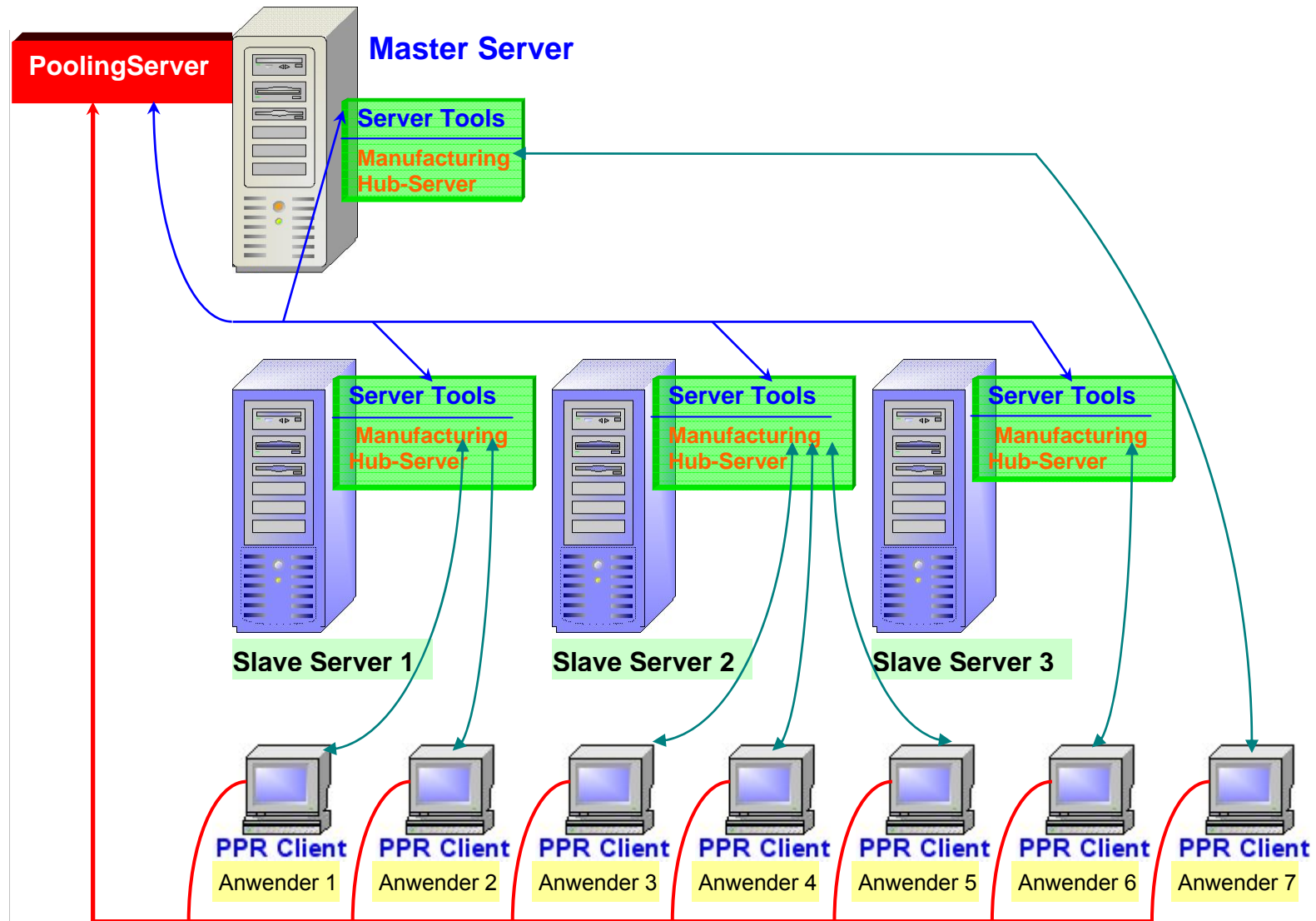


Abbildung 16: Client-Server Netzwerk

Arbeiten mit den Server Tools

Wenn Sie sich über **Connect PoolingServer** mit dem PoolingServer und damit mit dem Master Server verbunden haben, sehen Sie alle gestarteten Server (Abbildung 17 a).

Wenn Sie sich über **Connect ServerPool** angemeldet haben, müssen Sie sich mit den Servertools eines bestimmten Servers verbinden. Dies entspricht der Oberfläche, die Sie bereits aus den Server Tools der PE Version 5.10 kennen (Abbildung 17 b).

Server Name [PID_1568]

Show clients

Reload

Um die Clients die mit einem Server verbunden sind anzuzeigen, gehen Sie so vor:

- ➊ In Ihrem Browser sehen Sie den Server auf dem Sie sich angemeldet haben. Öffnen Sie auf diesem Eintrag das Kontextmenü und wählen die Funktion **Show clients**.
- ➋ Mit einem Doppelklick auf den Server, oder mit einem nochmaligen Aufruf des Kontextmenüs und der Auswahl der Funktion **Reload**, werden Ihnen alle Clientdaten angezeigt.

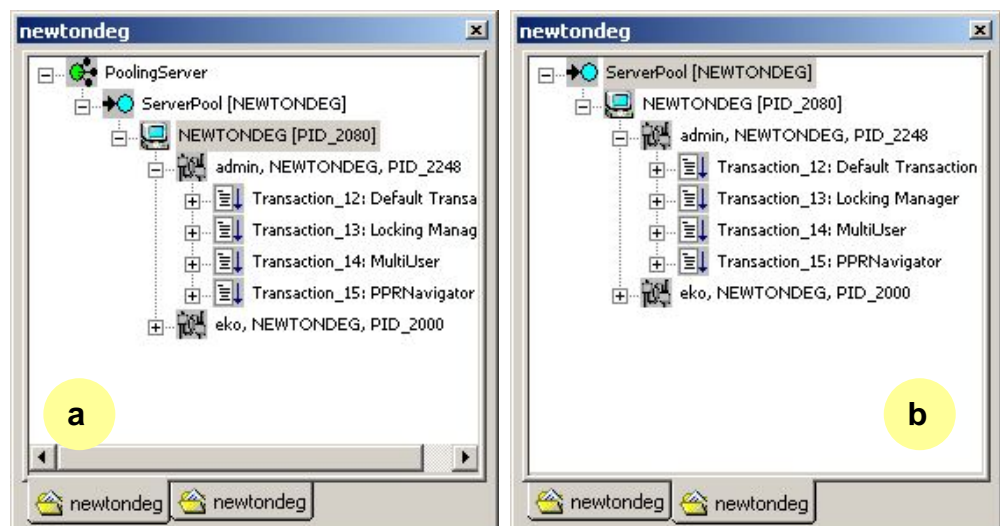


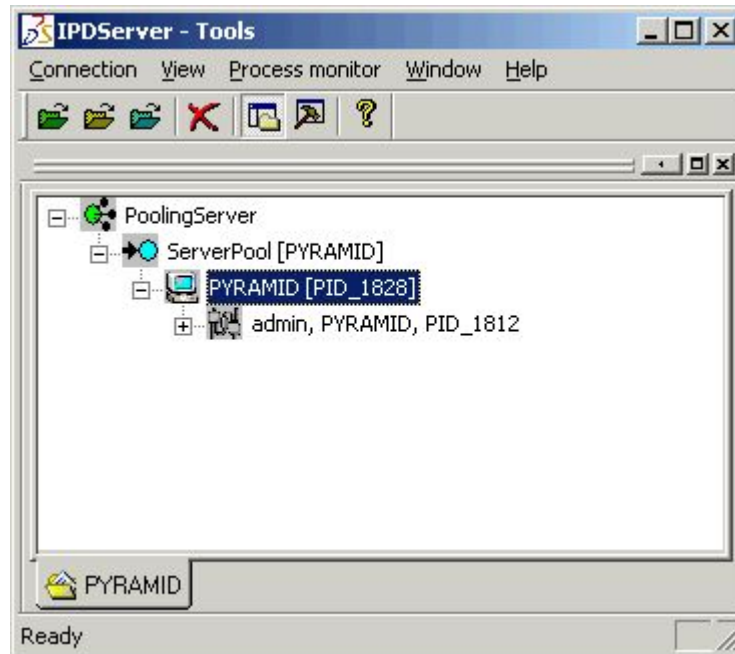
Abbildung 17: Server Tools mit 2 angemeldeten Clients

Ab der Version PE 5.14 kann man im ServerTools-Client erkennen, ob ein Manufacturing Hub-Server für neue Client-Anmeldungen gesperrt ist. Ein Manufacturing Hub-Server-Prozess kann für neue Clients gesperrt werden, wenn ein Fehler aufgetreten ist, der zu weiteren Folgefehlern führen kann (z. B.: Low Memory, Critical Server-Exception z. B. Access Violation, Critical Database-Exception bei Poet-Codes 6000-6199). In diesem Fall wird der betreffende Manufacturing Hub-Server in einen Emergency-Modus versetzt, in dem bereits verbundene Clients ihre Transaktionen abschließen können, neue Clients sich jedoch nicht mehr zu diesem Prozess verbinden können.

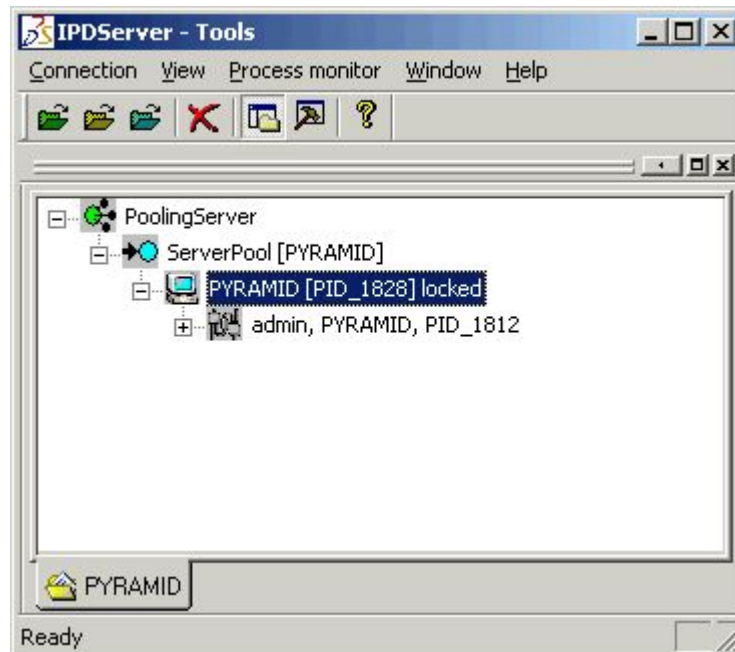
Beispiel

Nachfolgend werden die zwei Zustände des Manufacturing Hub-Server-Prozess 1828 auf PYRAMID gezeigt:

- nicht gesperrter Manufacturing Hub-Server-Prozess 1828 auf PYRAMID:



- Für eine neue PPR-Client Verbindungen gesperrter Manufacturing Hub-Server-Prozess 1828 auf PYRAMID:



- ⇒ der Sperrzustand wird beim Aktualisieren ('Reload') des Maschinen-Knotens angezeigt.

Das Kontextmenü des Servers



Abbildung 18: Kontextmenü der Server Tools

Server Einstellungen (Settings...)

Mit Hilfe dieser Einstellungen haben Sie immer direkten Zugriff auf den Registrierungseditor.

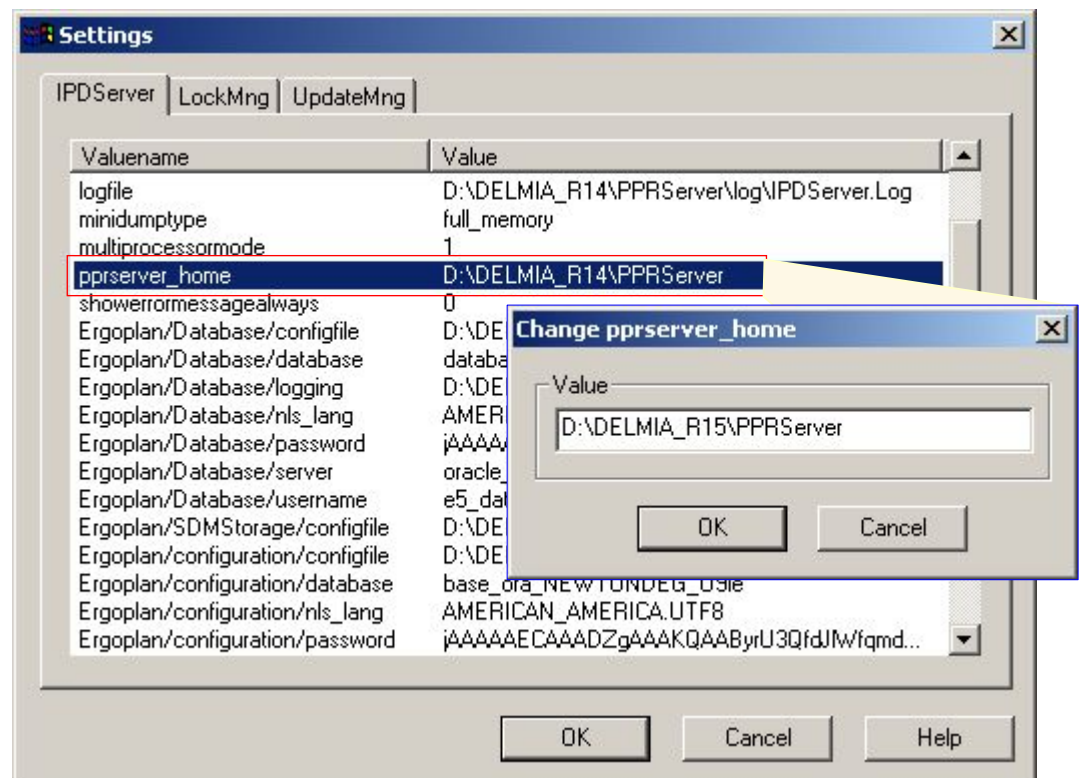


Abbildung 19: Die Server-Registrierungseinstellungen

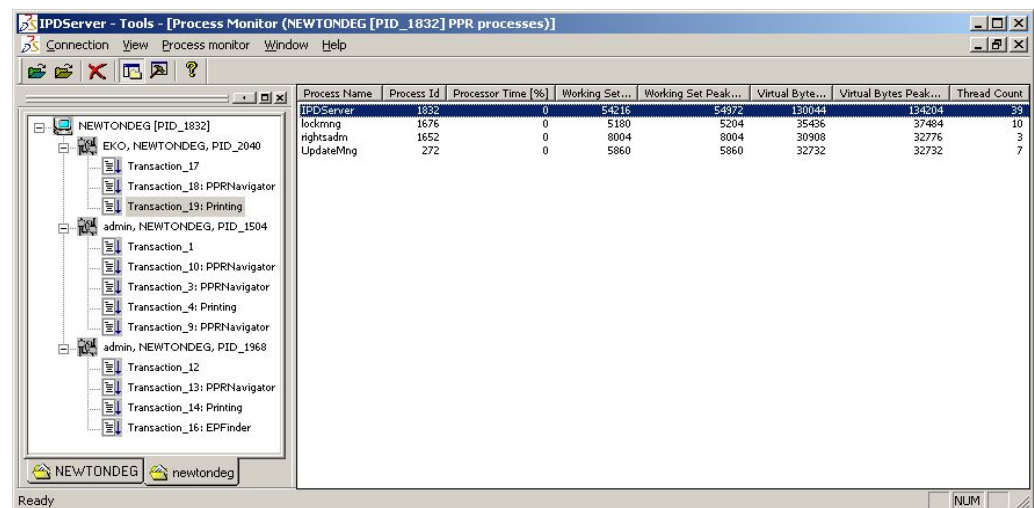


Abbildung 20: ein Master-Server ist gestartet (4 Serverprozesse), es sind 3 Clients verbunden

Terminate IPD server process...

Hiermit beenden Sie alle Serverprozesse.

Terminate clients...

Hiermit beenden Sie alle Clients, die mit dem Server verbunden sind.

Free Process Memory

Hiermit wird fragmentierter Freispeicher des Prozessheaps freigegeben. Der Prozessheap ist ein Teil des zur Verfügung stehenden Prozessspeichers.

Track Process Memory

Hiermit können die Anteile des Workingsets des Prozessspeichers, die sich im Vergleich zum letzten Aufruf dieser Funktion geändert haben, angezeigt werden. Es werden 2 Fenster angezeigt:

- neu allozierter Speicher
- aus dem Workingset entfernter Speicher.

Im Fenster „allozierter Prozess-Speicher“ wird auch der zum Laden von DLLs neu benötigte Speicher angezeigt.



Address	Size [KB]	Owner
007BE000	40	VirtualAlloc/Stack/Other
0204D000	20	D:\DELMIA\program\bin\ErgoplanDOImpl.dll
05141000	4996	VirtualAlloc/Stack/Other
05C68000	8696	VirtualAlloc/Stack/Other
090B0000	24	D:\DELMIA\PPRCClient\program\bin\GraphicDataps.dll
093BE000	8	VirtualAlloc/Stack/Other
094BE000	8	VirtualAlloc/Stack/Other
095BE000	8	VirtualAlloc/Stack/Other
096BE000	8	VirtualAlloc/Stack/Other
0C24C000	16	VirtualAlloc/Stack/Other
0C44B000	20	VirtualAlloc/Stack/Other
0C450000	828	VirtualAlloc/Stack/Other
0D74E000	8	VirtualAlloc/Stack/Other
0DB4E000	8	VirtualAlloc/Stack/Other
0DD4E000	8	VirtualAlloc/Stack/Other
0DE4E000	8	VirtualAlloc/Stack/Other
0DF4E000	8	VirtualAlloc/Stack/Other
0E04E000	8	VirtualAlloc/Stack/Other
0E14E000	8	VirtualAlloc/Stack/Other
0E24E000	8	VirtualAlloc/Stack/Other
0E34E000	8	VirtualAlloc/Stack/Other
0E44E000	8	VirtualAlloc/Stack/Other
48072000	16	D:\DELMIA\program\bin\pt6kn60.dll
4B00C000	4	D:\DELMIA\program\bin\pt6tn60.dll
4FC13000	12	D:\DELMIA\program\bin\pt6sn60.dll
77AA6000	8	C:\WINNT\system32\ole32.dll
77D56000	16	C:\WINNT\system32\RPCRT4.DLL
78027000	4	C:\WINNT\system32\MSVCRT.dll
7FF8C000	4	VirtualAlloc/Stack/Other
7FF8D000	4	VirtualAlloc/Stack/Other
7FF8E000	4	VirtualAlloc/Stack/Other
7FF90000	4	VirtualAlloc/Stack/Other
7FF91000	4	VirtualAlloc/Stack/Other
7FF92000	4	VirtualAlloc/Stack/Other
7FF93000	4	VirtualAlloc/Stack/Other
7FF94000	4	VirtualAlloc/Stack/Other
7FFA0000	4	VirtualAlloc/Stack/Other
7FFA3000	4	VirtualAlloc/Stack/Other
7FFA7000	4	VirtualAlloc/Stack/Other
7FFDD000	64	VirtualAlloc/Stack/Other

Abbildung 21: Track Process Memory

Show clients oder Reload

Die Clientdaten werden nicht automatisch aktualisiert. Soll der aktuelle Stand angezeigt werden, führen Sie Reload aus.

Über diesen Kontextmenüeintrag werden Ihnen alle an diesem Server angemeldeten Clients angezeigt.

Registrierungseditor Einträge

Zusätzlicher Registrierungseditor Eintrag: EPLOGGER

Der EPlogger ist ein Serverprozess, auf den nicht direkt vom Client zugegriffen wird. Die folgende Architektur wird verwendet:



Der EPLogger Prozess läuft zentral auf dem Server, aber er kann auch auf jeder Servermaschine laufen. Mit welcher Maschine ein Manufacturing Hub Server sich verbinden soll, kann über den Registrierungseditor festgelegt werden. Der Manufacturing Hub Server verbindet sich mit dem EPLogger Prozess mit Hilfe von COM (Component Object Model = ist eine proprietäre Technologie, um unter Windows Klassen aus DLLs zu exportieren, die von allen 32-Bit Microsoft-Betriebssystemen unterstützt wird).

Der EPLogger ist ein COM Prozess, der von dem Manufacturing Hub Server erzeugt wird, wenn Daten protokolliert werden müssen. Die Protokolle des EPLogger Prozesses sind XML Dateien. Der Pfad für die Protokolldateien ist im Registrierungseditor gespeichert. Der Name für die XML Dateien hat folgenden Aufbau:

- **PnO_YYYY-MM-DD.xml** (für P & O Änderungen) und
- **Export_YYYY-MM-DD.xml** (für Export Protokolle).

Die XML Protokolldateien werden erstellt, nachdem sie die Maximalgröße erreicht hat. Wenn nichts festgelegt ist, liegt der Grenzwert bei 30KB. Ab dieser Schwelle wird eine neue Datei erzeugt.

Der Protokollmechanismus startet den Server. Die Protokollierung ist asynchron. Dies bedeutet, dass das Protokollieren des EPLogger in einem separaten *thread* stattfindet. Dadurch ist der kleinste Leistungsaufwand sichergestellt.

Auf Manufacturing Hub Server Seite:

Der Manufacturing Hub Server ist standardmäßig deaktiviert. Um ihn zu aktivieren gehen Sie so vor:

HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\IPDServer\EPLogger\EnableLog:

- 0 (deaktiviert, default)
- 1 (nur P&O Änderungen werden Protokolliert)
- 2 (nur der Export wird Protokolliert)
- 3 (alles wird protokolliert)

Die Servermaschine mit welcher der Manufacturing Hub Server sich verbinden soll, wird mit folgenden Registrierungsschlüssels gesetzt:

HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\IPDServer\EPLogger\server.

Auf der EPLogger Seite:

- ☐ Der Pfad der Protokolldatei wird mit dem folgenden Registrierungsschlüssel gesetzt:
HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\EPLogger\LogPath.
- ☐ Die maximale Protokolldateigröße kann mit Hilfe der folgenden Registrierungseinstellungen festgelegt werden:
HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\EPLogger\MaxLogfileSize
Dieser Schlüssel ist eine Voraussetzung um neue Protokolldateien **pro Tag** zu erstellen. Der Standardwert beträgt „ 30 “ KB.
- ☐ Der Pfad von fehlerhaften Anmeldungen wird mit dem folgenden Registrierungsschlüssel gesetzt:
HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\EPLogger>ErrorLoggingPath
Beachten Sie bitte, dass nicht nur der Pfad sondern auch der Dateiname eingetragen werden muss, z. B.: C:\log\eperror.log.
- ☐ Die fehlerhafte Anmeldung wird mit Hilfe vom folgenden Registrierungsschlüssel freigegeben:
HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\EPLogger>ErrorLoggingEnabled
Jede Änderung, die in der Registrierung gemacht wurde, bleibt unwirksam bis der IPDServer und der EPLogger neu gestartet wird.

Einschränkungen

- Jede manuelle Änderung für die Protokolldatei, kann zu einem Verlust der Daten führen.
- Da die Daten asynchron zum EPLogger gesendet werden, und Fehler im EPLogger nach einer bestimmten Zeitspanne an den IPDServer und dann an den Client berichtet werden, kann es zu einem Verlust von Anmeldeinformationen kommen.

Zusammenfassung alle Einträge

HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\EPLOGGER

Registrierungseintrag	Wert	Erklärung
ErrorLoggingEnabled	'1'	Ein zusätzliches Fehlerprotokoll wird mit (1) eingeschaltet oder mit (0)ausgeschaltet. Aber selbst im ausgeschalteten Zustand werden schweren Fehlern protokolliert.
ErrorLoggingPath	'<install dir> \Log\eplogger.log'	Pfad der Error Logdatei.
failsafememory	'1024'	Größe in KB von dem reservierten „failsafe“ Speicher.
iobuffersize	'256'	Größe in KB von dem „iostream“ Puffer. Erst wenn dieser Wert überschritten wird, wird mit der Ausgabe begonnen.
LogPath	'<install dir> \Log\EPLogger\xml'	Verzeichnis um die Log Output xml Dateien zu speichern.

Registrierungseintrag	Wert	Erklärung
MaxLogFileSize	'10000'	Maximale Protokolldateigröße in KB.
memorylimit	'800'	Speicherlimit in MB für eine neue Manufacturing Hub-Server Verbindungen; wenn der EPLogger Prozess dieses Limit überschreitet, wird ein neuer Manufacturing Hub-Server zu einem anderen EPLogger Prozess verbunden.

Optionale Registrierungseinträge

Die Server-Installation fügt diesen Eintrag nicht in die Registrierung ein. Für LockMng.exe und UpdateMng.exe wird nach dem Setup der Standardwert 'normal' verwendet.

Bei schweren Fehlern, die eine genauere Dump-Analyse erfordern, sollte der Typ 'full_memory' in dem Registrierungseditor für den betreffenden Serverprozess gesetzt werden.

updatemng.exe

Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\Ergoplan UpdateMng
Eintrag: MiniDumpType
Wert: 'normal'

lockmng.exe

Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\Ergoplan LockMng
Eintrag: MiniDumpType
Wert: 'normal'

Mögliche Minidump-Typen:

Registrierungseintrag	Beschreibung
'none'	kein MiniDump wird geschrieben
'normal'	MiniDump mit Thread tack Daten
'with_data_segments'	Globale Datensegmenten von den geladenen Modulen
'full_memory'	MiniDump mit allen zusätzlichen Prozess Daten
'with_handle_data'	Zusätzliche Informationen zu OS handles.
'filter_memory'	Mit dem kompletten Speicher, aber das Löschen von ptr Werten wird nicht im Stack rekonstruiert.
'scan_memory'	Gesamter Speicher, ptr Referenzen werden gescannt.

cfcdatatransfer

Schlüssel: HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\ergoplan

Eintrag: cfcdatatransfer

Wert: '1'

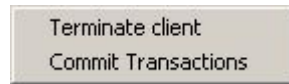
Mit dem Registrierungseintrag 'cfcdatatransfer' kann die Übertragung von Konfigurationsdaten, der ConfigFactoryCache, vom Server zum Client für 3 Szenarien optimiert werden:

- Client u. Server im LAN:
der Speicherbedarf (Größe des virtuellen Adressraums) für den Client-prozess liegt deutlich unter 2 GB ► - Wert '**1**'
- Client u. Server im LAN:
der Speicherbedarf (Größe des virtuellen Adressraums) für den Client-prozess kann 2 GB erreichen ► - Wert '**0**'
- Client u. Server im WAN: ► - Wert '**1**'

Mit dem **Wert '1'** werden alle Attribute der Konfigurationstypen mit dem Typ in einem Aufruf vom Server zum Client übertragen. Mit dieser Einstellung werden die Serverantwortzeiten insbesondere im WAN für erstmalige Konfigurationszugriffe optimiert.

Wird der **Wert '0'** verwendet, werden Attribute nur bei Bedarf vom Server geladen. Da die Einstellung auch für Planungstypen wirksam ist, kann bei einer hohen Anzahl von im Client verwendeten Planungstypen der Speicherbedarf des Clients reduziert werden (Speicherbedarf kann 2 GB erreichen), wenn nicht auf alle Attribute und Attributtypen dieser Planungstypen zugegriffen wird.

Das Kontextmenü auf dem Client



Terminate client

Hiermit beenden Sie den Clientprozess.

Commit Transaktion

Wenn ein Client nicht mehr bedient werden kann (er bekommt keine Antwort vom Server), können die noch nicht gespeicherten Daten (offene Transaktionen) hiermit gesichert werden.

Beenden von Clients

Mit der Einführung des PoolingServer in der Version PE 5.11, wurde die Serverinfrastruktur des DELMIA Process Engineers komplexer. In den nachfolgenden Versionen ist der Funktionsumfang der Gleiche geblieben, nur die Verwaltung des PoolingServers wurde erweitert.

Im Kontextmenü des PoolingServer und des ServerPools finden Sie zwei Einträge. Beide beziehen sich auf das Beenden von Clients. In diesem Abschnitt wird diese Funktionserweiterung näher beschrieben.

Vorgehensweise

Wo können Clients beendet werden?

Die Beendigung eines Clients kann von folgenden Knoten gestartet werden:

- **PoolingServer.** Auf dem PoolingServer werden **alle** Clients der DELMIA Process Engineer® Installation beendet.
- **ServerPools.** Auf den ServerPools werden **alle** Clients der selektierten Servermaschine beendet.
- **Servermaschine.** Alle Clients dieses Manufacturing Hub-Server Prozesses werden beendet.
- Auf dem **Client** selbst. Nur dieser Client wird beendet.

Was geschieht beim Beenden von Clients?

Beim Abschalten von Clients müssen zwei Standpunkte betrachtet werden.

- ⇒ Der Standpunkt des Administrators der die Clients beendet.
- ⇒ Der Standpunkt des Anwenders dessen Client beendet wird.

Beendet der Administrator einen oder mehrere Clients, erscheint

- ⇒ ein Dialog in dem zwei Zeitangaben möglich sind. Die Standardwerte sind jeweils 10 Minuten.

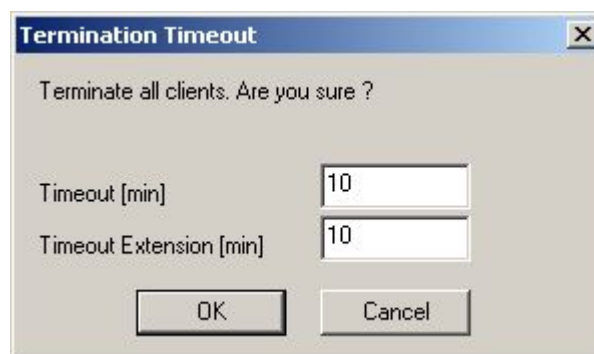


Abbildung 22: Termination Timeout

- Mit der Zeitangabe *Timeout* legt der Administrator die Zeitspanne fest, nach der der Client beendet werden soll. Wird hier als Wert 0 eingetragen, wird die sofortige Beendigung des Clients gestartet, wobei offene Clienttransaktionen automatisch gespeichert werden. Der Clientbenutzer kann innerhalb von 30 Sekunden optional die Anwendung selbst beenden und dabei Entscheiden ob die letzten Änderungen gespeichert werden sollen oder nicht.
Wird ein Wert > 0 eingegeben, kann der Clientbenutzer zwischen mehreren Terminierungsmodi für das Speichern von Transaktionen wählen (siehe unten).
 - Die zweite Zeitangabe (*Timeout Extension*) gibt die Zeitspanne an, die der Anwender zur Verfügung hat, um das Beenden zu verzögern. D. h. die Zeitdauer die hier angegeben wird, kann der Anwender benutzen um die Clientterminierung zu verlängern. Legt man die Standardwerte zugrunde, hat der Anwender die Möglichkeit das Beenden des Clients auf 20 [min] zu verlängern. Wird der zweite Zeitwert auf 0 gesetzt, wird nach 10 Minuten der Client beendet und der Clientanwender hat keine Möglichkeit die Beendungsdauer zu beeinflussen.
- ⇒ Bei allen Anwendern, deren Clients beendet werden, erscheint ein Dialog mit dem der Terminierungsmodus und Zeitdauer bis zur Terminierung beeinflusst werden können.

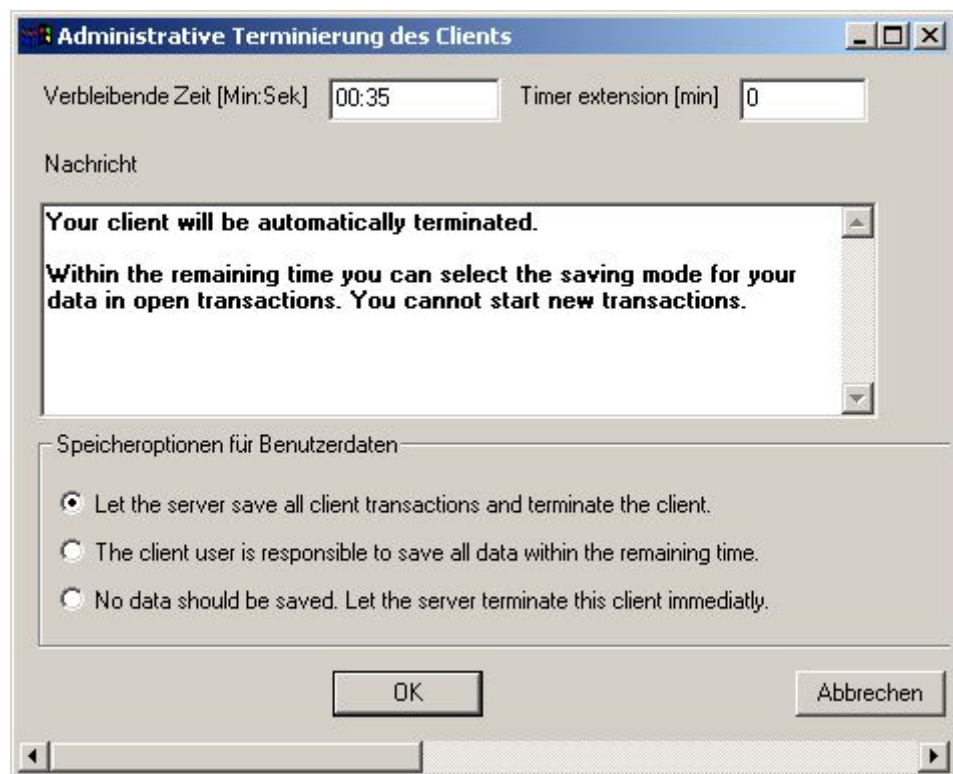


Abbildung 23: Hinweis auf das Beenden des Clients

Die verbleibende Zeit bis zur Terminierung des Clients wird im oberen Teil des Dialogs angezeigt. Die verbleibende Zeit kann durch der Anzahl von Minuten im Feld *Timer extension* verlängert werden. Dies ist nur dann möglich, wenn der Administrator eine *Timer extension* zugelassen hat. Ist dies der Fall, wird

die verbleibende Zeit um das Maß erhöht, dass in dem Feld *Timer extension* eingetragen und über den Button *Anwenden* aktiviert wurde. Ist der Button *Anwenden* nicht sichtbar, muss der Dialog nach links vergrößert werden.



Hinweis:

Wenn Sie nicht rechtzeitig den Beendungsprozesses verlängern, erscheint nachfolgender Meldung:



Nur die Einträge werden berücksichtigt, die in der zuvor angezeigten Restzeit vom Server verarbeitet werden.

Im unteren Teil des Dialogs legen Sie die Speicheroptionen fest.

- Der Server sichert alle offenen Transaktionen
- Der Clientbenutzer ist für das Sichern der Daten verantwortlich, der Server sichert keine Transaktionen.
- Keine Daten sollen gesichert werden. Der Server sichert keine Transaktionen.

Der Benutzer kann den Dialog schließen - dies hält die Beendigung nicht an.

Die Speicheroptionen sowie die Zeiten können durch das Betätigen des Buttons *Anwenden* gewechselt werden. Um die Auswahl zu aktivieren, muss der Dialog nicht geschlossen werden. Wurde der Dialog geschlossen, kann er vom Clientbenutzer nicht wieder geöffnet werden. Es werden die zuletzt gewählten Einstellungen verwendet.

Während der Beendungsdauer können Sie den Dialog schließen oder geöffnet lassen.

⇒ Ist die verbleibende Zeit abgelaufen, wird nochmals ein Hinweisfenster für 30 Sekunden eingeblendet.

War der Eingabedialog noch geöffnet, ändert sich nur der Text in dem Ausgabefenster – die gewählte Speicheroption wird angezeigt.

Die Speicheroptionen können jetzt nicht mehr geändert werden, das Speichern von Transaktionen ist aber noch möglich.

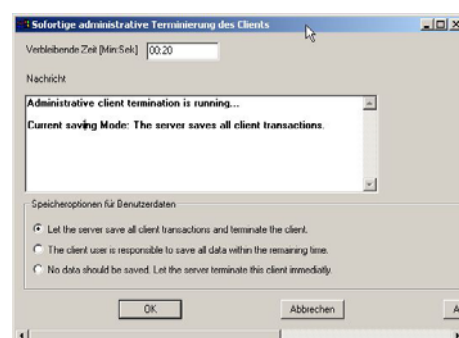


Abbildung 24: Hinweis auf das Beenden des Clients

- ⇒ Nach Ablauf der 30 Sekunden, erscheint ein Meldungsfenster mit dem Hinweis, dass die Verbindung zum Server beendet wurde. Jetzt besteht keine Speichermöglichkeit mehr, der Dialog kann nur mit OK oder Abbrechen beendet werden. In beiden Fällen wird die Anwendung geschlossen.



Abbildung 25: Meldung (an den Clientbenutzer) dass der Client beendet wurde

- ⇒ Der Administrator erhält nun die Meldung über das Beenden des Clients. Die Meldung erscheint für jeden ServerPool gesondert, wenn die Client-terminierung über den PoolingServer-Knoten gestartet wurde.



Abbildung 26: Server Tools Meldung (an den Administrator) nach dem Beenden eines Clients

EPUnlockTool

Allgemein

Was ist das EPUnlockTool?

Bisher konnten dauerhafte Datensperren (gelockte Daten, werden im weiteren Verlauf als persistente *Locks* bezeichnet) nur mit Hilfe der Server Tools aufgehoben werden. Die Server Tools sind ein Administrationstool und nicht für jeden Anwender zugänglich. Dauerhafte Datensperren für einen bestimmten Benutzer müssen gelöscht werden, wenn diese Daten nicht mehr bearbeitet werden und anderen Benutzern schreibenden Zugriff ermöglicht werden soll. Das gilt auch für den Benutzer, der die Datensperre ausgelöst hat, wenn die Anwendung abgestürzt oder die Serververbindung ausgefallen ist und der Client neu zum Server verbunden werden muss. Permanente Datensperren sind zur Serverlaufzeit wirksam, d. h. wird der PPRServer heruntergefahren, werden diese Locks auch aufgehoben.

Mit dem PPRClient Tool „**EPUnlockTool**“ erhalten Sie die Möglichkeit, inaktive persistente Locks z. B. nach einem Client-Crash, freizugeben. Damit können zuvor gesperrte Daten wieder bearbeitet werden. Dies trifft vor allem auf V5 Anwender zu, da nach einem Absturz nur der Systemadministrator gelockte Daten wieder freigeben konnte. Mit der Benutzung des neuen Client Tools kann der Anwender seine dauerhaften Locks selbst freigeben und damit auch sich selbst und andere Benutzer diese Daten zugänglich machen.

Das EPUnlockTool ist ein Clientprogramm ohne eigene Benutzeroberfläche. Fehlermeldungen und andere Informationen werden in Message-Dialogen ausgegeben. Fehler werden zusätzlich in das LogFile "EPUnlockTool.log" im Client-Log-Verzeichnis geschrieben.

Funktionsweise

- Der Client Benutzer muss sich mit dem Benutzernamen und dem Passwort anmelden. Die Anmeldung erfolgt mit der gewohnten PPR Client Anmeldungs-Funktionalität.
- Der Benutzer bestätigt das Löschen der im zugeordneten persistenten Locks in einem Dialog. Danach werden inaktive persistente Locks, die dem Benutzer und dem verwendeten Rechner zugeordnet sind, gelöscht.
- Aktive, dauerhafte Datensperren die mit offenen Transaktionen verbunden sind, können **nicht** gelöscht werden. Um die Datenintegrität innerhalb des DPE zu gewährleisten, ist es wichtig, dass der Client Anwender diese aktiven Locks nicht löschen kann.
- Wenn der Client abstürzt, so werden seine Transaktionen vom DCOM Cabbage Collector innerhalb von 6 Minuten geschlossen. Dauerhafte Locks können nur nach dieser Zeitspanne mit dem „EPUnlockTool“ für den abgestürzten Client gelöscht werden.
- Wenn das Löschen fehlschlägt, wird eine Fehlermeldung angezeigt.

- Alle Fehler werden in der Logfile "EPUnlockTool.log" gespeichert und befinden sich in der PPR Client Log Verzeichnis.

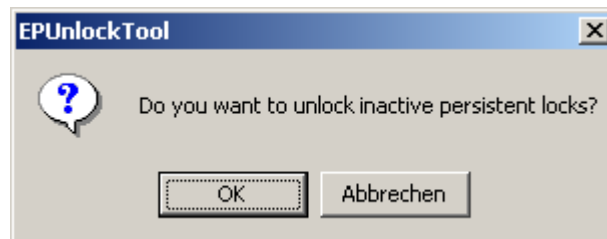
Bedienungsanleitung



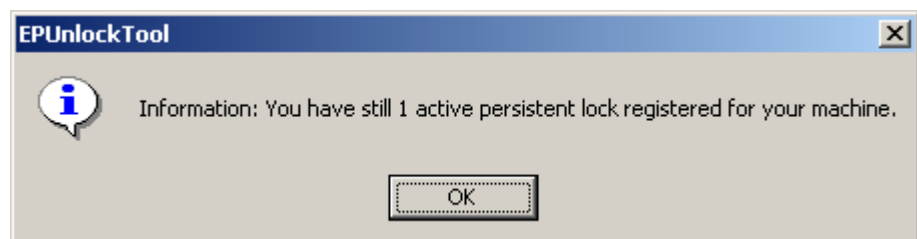
- ➔ Öffnen Sie das Verzeichnis Ihrer DELMIA Process Engineer® Installation. Unter `..\DELMIA\PPRClient\program\bin` finden Sie die ausführbare Datei ***epunlocktool.exe***.
- ➔ Starten Sie die Datei ***epunlocktool.exe*** um Sperrungen aufzuheben.
- ⇒ Es öffnet sich der DELMIA Process Engineer® Anmeldedialog



- ➔ Nach der Anmeldung erscheint der einzige interaktive Dialog.



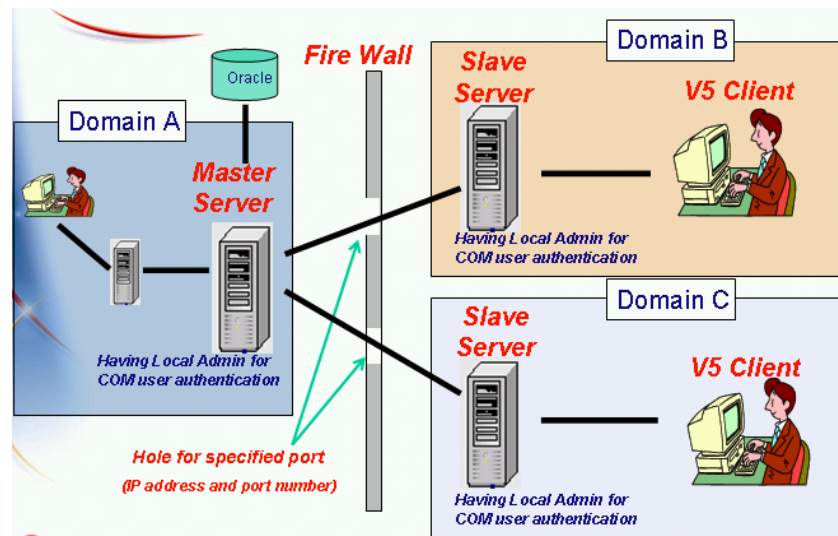
- Wenn der Button OK gedrückt wird, werden alle inaktive, dauerhafte Locks für den angemeldeten Benutzer und die Maschine, gelöscht.
- Wenn der Button „Abbrechen“ gedrückt wird, werden keine dauerhaften Locks gelöscht.
- ⇒ Es werden alle inaktiven, dauerhaften Sperrungen freigegeben.
- ⇒ Wurden nicht alle inaktiven dauerhaften Sperrungen freigegeben, wird in einem Informationsdialog die Anzahl der verbleibenden, für den Benutzer an der gegenwärtigen Maschine, aktiven Locks angezeigt.



Freigabe von persistenten Sperrungen durch einen Slave Server

Ausgangssituation

Es existieren mehrere Domänen mit einer Firewall zwischen den einzelnen Domänen in einem Netzwerk. Die Kommunikation die über die Firewall hinausgeht, ist sehr begrenzt und erlaubt nur den Slave Server Maschinen eine Verbindung zu der Master Server Maschine. Die Clients sind in derselben Domäne wie der ihnen zugewiesene Slave Maschine. Da der Zugang auf den Master Server über die Firewall für alle Clients blockiert ist, kann das EPUnlockTool nicht ausgeführt werden.



Um die vorhin beschriebenen Einschränkungen zu überwinden, wurde eine API für den Manufacturing Hub-Server Prozess eingeführt. Das EPUnlockTool verwendet diese API, anstatt auf den Lockmanager des Masterserver direkt zuzugreifen. Die Implementierung der neuen API Methode im Manufacturing Hub-Server erlaubt es dem Master Lockmanager auch bei einer Firewall die Unlock Funktionsweise fortzusetzen. Da dem Manufacturing Hub-Server Prozess ein Zugriff auf den Master Lockmanager in beiden Zugangsszenarien (eine von der Sklave Server durch die Firewall oder auf der Master Maschine direkt) erlaubt, steht das EPUnlockTool allen Clientmaschinen in der Clientumgebung zur Verfügung. Die Funktionalität des EPUnlockTool wird nicht verändert. Das EPUnlockTool verhält sich mit dem beschriebenen Umfeld wie in einem Umfeld, dass keinen Firewall benutzt.

Vorraussetzungen und Einschränkungen

Die Firewall muss konfiguriert werden um eine DCOM Kommunikation zwischen Master und Slave Server explizit zu erlauben. Die Kommunikation zwischen Master Server und Client Maschine muss gesperrt sein.

Einschränkung

Die Funktionalität vom EPUnlockTool wird nur in einer Umgebung die mit einer Firewall geschützt ist, zur Verfügung stehen, wenn die Firewall die DCOM Kommunikation zwischen Slave und Master Server erlaubt. Bei Benutzung einer separaten Datenbank Maschine, muss die Kommunikation zwischen allen DPE5 Server Maschinen und der Datenbank Maschine erlaubt werden.

Abbildungsverzeichnis

Abbildung 1: Schema PoolingServer – Server Tools und Clients	12
Abbildung 2: Welche Prozesse laufen wo?	14
Abbildung 3: Poolingserver-Konfigurationsdatei	17
Abbildung 4: Server Tools über das Startmenü starten	24
Abbildung 5: Oberfläche beim Start der Server Tools	24
Abbildung 6: Dialogbaustein <i>Connect <u>P</u>oolingServer</i> , <i>Connect <u>S</u>erverPool</i> und <i>Connect <u>L</u>ockMng</i>	25
Abbildung 7: Dialogbaustein <i>Lock Information</i> in dem alle gelockten V5 Transaktionen aufgeführt werden	26
Abbildung 8: PoolingServer und Server Tools mit 2 angemeldeten Clients	28
Abbildung 9: Kontextmenü des PoolingServer	29
Abbildung 10: Eigenschaften des PoolingServers	29
Abbildung 11: Eigenschaften des PoolingServers Administrations-Clients	30
Abbildung 12: „Show ServerPool load indexes“ des Pooling Servers	34
Abbildung 13: Kontextmenü des ServerPools	38
Abbildung 14: Eigenschaften des ServerPools	39
Abbildung 15: Show Process Information	43
Abbildung 16: Client-Server Netzwerk	46
Abbildung 17: Server Tools mit 2 angemeldeten Clients	47
Abbildung 18: Kontextmenü der Server Tools	49
Abbildung 19: Die Server-Registrierungseinstellungen	49
Abbildung 20: ein Master-Server ist gestartet (4 Serverprozesse), es sind 3 Clients verbunden	50
Abbildung 21: Track Process Memory	51
Abbildung 22: Termination Timeout	58
Abbildung 23: Hinweis auf das Beenden des Clients	59
Abbildung 24: Hinweis auf das Beenden des Clients	60
Abbildung 25: Meldung (an den Clientbenutzer) dass der Client beendet wurde	61
Abbildung 26: Server Tools Meldung (an den Administrator) nach dem Beenden eines Clients	61

Index

A

Arbeitsspeicher	70
Auslagerungsdateien	71

B

Beenden von Clients	58
Vorgehensweise	58
Button <i>Anwenden</i>	60

C

Client Kontextmenü	
Commit Transaktion	57
Terminate client	57
<i>Connect PoolingServer</i>	47
cpuload_weight	30

E

EPlogger	53
EPUnlockTool	62
Bedienungsanleitung	64
Funktionsweise	62
Persistente Locks	65

F

feememory_weight	30
Firewall	65

H

Haftungsausschluss	3
Handles	69
Heap	69

I

integration_intervall	31
-----------------------------	----

L

Lastbalancing	33
load_distance_threshold	33
Lastdaten	70

Lastindex	70
Lastkriterien	30
Lastverteilung	70
load_distance_threshold	32
loadgradient_intervall	31
Loadgradient_weight	30
Lock PoolingServer	35
locked_on_start	32

M

Machinepriority_weight	30
Master Server	11, 46
measurement_intervall	31
Memorylimitconvergence_weight	30
Memorypagefault_weight	30
Memoryusage_weight	30
Menü Help	27
Menü <i>Process monitor</i>	27
Show all Process	27
Update speed	27
Menü <i>View</i>	27
Menü <i>Window</i>	27
Menüleiste	27
Messintervall	
Integration_intervall	34
Loadgradient_interval	34
verändern	34

N

Neue Funktionen:	9
------------------------	---

P

PoolingServer	11
PoolingServer Kontextmenü	
Settings	29
Show ServerPool load indexes	34
Terminate processes	36
Poolingserver-Konfigurationsdatei	16, 69
Process ID	70

R

RAM.....	70
Registrierungs- Editor.....	16
Registrierungseditor	
EPlogger	53
Optionale Registrierungseinträge	55
Reload.....	28, 44, 47, 52

S

Server Kontextmenü	
Settings... ..	49
Server Tools	
clientlimit	41
Starten und Beenden	24
Connect <u>P</u> oolingServer	25
Connect <u>S</u> erverPool	25
Servernet	70
Serverpool	41
memorylimit	41, 42
serverpoolresponse_intervall	31
Serverpoolresponse_weight	30
ServerPools	
Kontextmenü	
Settings	38

Settings	29
Show clients	47, 52
Show running ServerPools	28
Show Server Instances	28
Show ServerPool load indexes	34
Speicherseitenfehler	70
Speicherseitenfehlern	
Memorypagefault_weight	34
Subnet	71

T

Terminate clients	35, 36
Terminate processes	36
Terminate unused IPD server processes... ..	35
Thread	71
Timeout	59
Timeout Extension	59

V

Virtueller Speicher.....	71
--------------------------	----

W

Werkzeugleiste	27
Working Set Peak	71

Anhang

Begriffsdefinitionen

Administrations- Clients:

Administration Clients sind Client Module, die für bestimmte Aufgaben, zum Zwecke der Administration, verwendet werden. Diese Clients benötigen spezielle Zugänge zu der E5 Datenbank und den Servervorgängen. Administrations- Clients, die für bestimmte Anwendungen in der Datenbank verwendet werden, wie z. B. ein Datenimport, haben spezielle Anforderungen an die Systemressourcen (Datenspeicher, CPU) welche für diese Aufgaben reserviert werden sollten. Beispiele von E5 Administrations- Clients: pprloader.exe, dbanalyser.exe, ptimex.exe

PoolingServer:

Prozess auf der Master Maschine, der die Performancedaten der Server Maschinen misst.

Poolingserver-Konfigurationsdatei:

Mit Hilfe der Poolingserver-Konfigurationsdatei können Sie den PoolingServer steuern, bzw. können Sie eine Strukturierung Ihres Client-Server Netzwerkes vornehmen. Der Registrierungs-Schlüssel der Konfigurationsdatei ist:

...Delmia\Ergoplan PoolingServer\ConfigFile value: <Filepath>

ServerPool:

Misst die Performancedaten der einzelnen Prozesse auf einer (seiner) Maschine.

Server Tools:

Die ServerTools messen die Performance der einzelnen Prozesse eines Clients.

IPD-Prozesse:

IPD-Prozesse sind die Prozesse die auf der Integrierten Prozess Datenbank des DELMIA Process Engineers ablaufen.

Client net:

Ist die Zuordnung von Clientgruppen zu den Serverdomänen.

Handles:

Ein Wert, der eine Ressource, beispielsweise eine Datei oder einen Registrierungsschlüssel, eindeutig identifiziert, so dass ein Programm auf diese Ressource zugreifen kann.

Heap:

Ist ein Teil der Windowsspeicherverwaltung für die Zuteilung kleiner Speicherblöcke.

Lastdaten:

Sind alle für die PPRServer Lastindexe herangezogenen Performance-Daten. Die Daten fallen einerseits auf der Maschine selbst, anderseits von einem ausgewählten Manufacturing Hub-Serverprozess an.

Lastindex:

Ist eine Maßzahl für die PPRServer Belastung. Je höher der Lastindex, um so größer die Belastung.

Lastverteilung:

ist die möglichst gleichmäßige Auslastung von verfügbaren PPR-Server-Instanzen. Die Lastverteilung des **PoolingServer** bezieht sich nur auf die Zuteilung von neuen Clients.

Process ID:

Ist die eindeutige Nummer unter der der Prozess geführt wird.

Ein numerischer Bezeichner, mit dem ein Prozess während seiner Ausführung eindeutig gekennzeichnet ist. Sie können PIDs im Task-Manager anzeigen lassen. PID steht für *Process Identifier*.

Prozess: = der Betriebssystem-Prozess

Wenn in diesem Handbuch des Öfteren von ‚Prozessen‘ die Rede sein wird, so sind damit nicht die Prozesse aus der Prozesssicht des DELMIA Process Engineers gemeint, sondern es sind die Betriebssystemprozesse.

RAM oder Arbeitsspeicher:

Speicher, in den ein Computer Informationen schreibt und / oder ausliest. Im RAM gespeicherte Informationen gehen beim Ausschalten des Computers verloren. RAM steht für Random Access Memory (Arbeitsspeicher). Reicht der Arbeitsspeicher für das Ausführen eines Programms nicht aus, muss ein zusätzlicher Speicher („Virtueller Speicher“) verwendet werden.

Serverdomänen:

Für den PoolingServer ist eine Serverdomäne eine Gruppe von Serverrechnern.

Servernet:

Ein Servernet besteht aus allen Serverdomänen.

Speichernutzung:

Ist das aktuelle Workingset eines Prozesses in Kilobyte. Das aktuelle Workingset ist die Größe aller aktuell für einen Prozess in den Speicher geladenen Seiten. Die Seiten finden Sie im Task-Manager unter dem Reiter Prozesse unter der Spaltenüberschrift Speichernutzung.

**Speicherseitenfehler:**

Ein Speicherseitenfehler wird ausgelöst, wenn eine Applikation auf eine Speicherseite des Workingsets im Hauptspeicher zugreifen will und diese nicht mehr eingetragen ist. In den meisten Fällen folgt einem Speicherseitenfehler ein Zugriff auf die Auslagerungsdatei oder eines Speicherseitencaches.

Im Task-Manager gibt der Seitenfehler die absolute Anzahl der Lesevorgänge an. Der Seitenfehlerwert erhöht sich ab dem Zeitpunkt des Prozessaufufes.

Der PoolingServer berücksichtigt die Speicherseitenfehlerrate pro Sekunde.

Subnet:

Gedachter Teilbereich eines bestehenden Client-Server Netzwerkes.

Thread :

Ein Programmausführungspfad innerhalb eines Prozesses. Threads erlauben gleichzeitige Vorgänge innerhalb eines Prozesses und ermöglichen einem Prozess, verschiedene Teile seines Programms auf unterschiedlichen Prozessoren gleichzeitig auszuführen.

Angezeigt wird die Anzahl der Threads.

Virtueller Speicher oder Virtual Bytes:

Temporärer Speicher, der vom einem Computer zum Ausführen von Programmen verwendet wird, die mehr Speicher benötigen, als tatsächlich vorhanden ist. Programme können beispielsweise auf bis zu 2 GB virtuellen Speicher auf der Festplatte des Computers zugreifen, selbst wenn der Computer eigentlich nur über 128 MB RAM verfügt. Die Programmdateien, die aktuell nicht in den RAM-Speicher des Computers passen, werden in *Auslagerungsdateien* gespeichert.

Working Set Peak:

Höchstwert des Virtuellen Speichers. Ist der vom Prozess aktuell genutzte Adressraum des Speichers.

Historie: Verbesserungen im Server Pooling

DPE5.12

Server Tools Client:

- ☐ **Änderung der administrativen Beendigung von E5-Clients (DPFFrame nur Clients):**
 - Konfigurierbare Beendigung vom Timeout.
 - Automatischer Speicher vom Arbeitsvorgang (es wird ein Commit ausgeführt).
 - Das Beenden läuft in einem eigenen Thread.
- ☐ **Neuer Menüeintrag auf dem PoolingServer Knoten: Lock Pooling-Server**
 - Eine neue Clientverbindungen ist nicht möglich.
 - Erlaubt die Durchführung von Administrationsaufgaben (außerhalb von DPE/DPM).

DPE5.13

Server Tools Client:

- ☐ **Neues Menü: *terminate unused Manufacturing Hub-Server processes***
 - ⇒ Beendet die Manufacturing Hub-Server-Prozesse, zu denen keine PE-Client mehr verbunden sind.
 - ⇒ Der Batch Befehl lautet: /gc_Manufacturing Hub-Server

PoolingServer:

- ☐ Die Lastmessung auf Basis von Windows-Performance-Indikatoren kann für eine PPR-Serverdomäne (Gruppe von PPR-Servermaschinen) in der PoolingServer-Konfiguration abgeschaltet werden

DPE5.14

ServerTools Client:

- ☐ Mit Connect LockMng verbinden Sie sich mit dem Lock-Manager. Es öffnet sich ein Dialog in dem die gelockten Clients aufgeführt werden. Hier haben Sie nun die Möglichkeit die gelockten Client freizugeben. Diese Funktionalität steht Ihnen nur in Verbindung mit V5 zur Verfügung.

Achtung: Mit dem externen 'unlock' tool, das **EPUnlockTool**, können ebenfalls gelockte Clients freigegeben werden.

☐ **Neue Batch Befehlszeilen:**

- /poolingserver_lock_on_start
- /poolingserver_unlock_on_start
- /poolingserver_lock
- /poolingserver_unlock

PoolingServer:

- ☐ Zur Speicherfreigabe werden automatisch ungenutzte IPD- Server Prozesse beendet.
- Die Untersuchung startet automatisch in einem konfigurierten Zeitintervall.
- ☐ Erweiterung der Konfigurationsdatei: Client – Maschinen, die nicht in der PoolingServer Konfiguration aufgelistet sind, können der Server Domäne '@anonymous_userdomain' zugewiesen werden.
- ☐ Erweiterung der Konfigurationsdatei: Client – Maschinen können der Domäne '@admin_domain' zugewiesen werden um auch dann eine Verbindung zu erlauben, wenn der Pooling Server gesperrt ist.
- ☐ Neuer Registrierungseintrag 'locked_on_start' erlaubt die Sperrung des Pooling Servers selbst wenn der PoolingServer Prozess neu gestartet wurde.
- Mit dieser Option können Administrationsvorgänge sicherer mit einer verbundenen gesperrten DPE Installation durchgeführt werden, selbst wenn die Installation neu gestartet werden muss oder gerade erst neu gestartet wurde.

DPE5.15

Server Tools Client:

☐ **Neue Batch Befehlszeilen:**

- /kill_all_pprserver_processes
- /kill_all_Manufacturing Hub-Server_processes

- ☐ **Der Prozess EPGenericServices wurde ebenfalls mittels 'terminate all pprserver processes' beendet.**

ServerTools:**❑ Neuer Registrierungseintrag 'pdh_init_timeout'**

- ⇒ Legt den Initialisierungs-Timeout für den Zugriff auf die Windows-PDH-Dll fest. Diese Dll wird zur Performance-Messung verwendet und kann z. B. bei hoher Rechnerauslastung das Starten des ServerPools über den COM-Initialisierungs-Timeout hinaus verzögern.
Wird die mit dem Registrierungseditor-Eintrag 'pdh_init_timeout' festgelegte Zeit bei dem Initialen PDH-Zugriff überschritten, wird die Performance Messung zur Lastbalancierung für diesen ServerPool ausgeschaltet.

ServerTools/PoolingServer:

- ⇒ Registrierungseinträge werden nur noch ohne die Verwendung von [] – Klammern erstellt.
- ⇒ Der PoolingServer und der ServerPool werden mit den SystemEnvironment Daten initialisiert, die im Registrierungseditor Eintrag: HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\Ergoplan System; gespeichert sind.
Diese Initialisierung ist zwingend notwendig um die PoolingServer und ServerPool Services von anderen Modulen zu nutzen.

DPE5.16**Server Tools Client:**

- ❑ Der Prozess EPLogger wird ebenfalls abgebrochen, wenn 'terminate all pprserver processes' durchgeführt wird.

PoolingServer:**❑ Es wurde eine Verbesserung der Kontrolle des RPC Status der COM Verbindungen zu dem Prozess der Server Tools eingeführt:**

- ⇒ Die Kommunikation zwischen den einzelnen Prozessen wird an zentraler Stelle verwaltet.
Jede PoolingServer Aufgabe kann den RPC Fehlerstatus aufrufen.

- ⇒ Die Reinitialisierung einer unterbrochenen RPC Verbindung wird mit der Meldung eines PoolingServer Reinitialisierungsereignisses unmittelbar nach dem ersten aufgetretenen RPC Fehler für die abgelegene COM Verbindung beschleunigt.

❑ Neuer Registrierungseintrag 'load_distance_threshold' erlaubt die Nutzung von einer Verbindungszählung basierend auf die Load Balancing Strategie, wenn die Ladedifferenz zwischen zwei Pool maschinen klein ist.

- ⇒ Der Eintrag 'load_distance_threshold' legt den Schwellwert der Lastdifferenz verfügbarer Servermaschinen fest, ab dem das Loadbalancing auf Basis von Maschinenlastmessungen angewendet wird.

❑ **Asynchrone Initialisierung vom ServerPooling Service: Lesen Sie Konfiguration und Verbindung von ServerPools**

❑ **Das automatische Wiederaufnahmeverfahren der Clientverbindung:**

- ⇒ Wenn die Clientverbindung aufgrund eines Fehlers der COM Instantiierung versagt hat, wiederholt sich die Clientverbindung nochmals, da ein COM Timeout eine Verbindung verhindern könnte, während der Manufacturing Hub-Server Prozess gestartet wurde.
- ⇒ Wenn während der Verbindung vom Client zum ServerPool ein ServerPool RPC Fehler auftritt, wird die Verbindung ein zweites Mal gestartet, da beim ersten Auftreten des Fehlers, der ServerPool durch die Administrationswerkzeuge beendet werden, infolge des aufgetretenen RPC Fehler.

Server Tools:

❑ **Um das Blockieren einer Clientverbindung vorzubeugen, Asynchrone Überprüfung der Manufacturing Hub-Server Gültigkeit wenn ein Manufacturing Hub-Server nicht antwortet.**

- ⇒ Mit dieser asynchronen Überprüfung wird das Multithreading Verhalten überprüft und der ServerPool optimiert. So werden schnellere Antwortzeiten erzielt, wenn Clients verbunden sind und/oder Pool Informationen vom Server Tools Client abgefragt werden.

DPE 5.16SP6

PoolingServer:

- ❑ Asynchrone Initialisierung des ServerTools Prozesses; Diese Initialisierung beinhaltet das Erzeugen des ServerPools und den Start der *load measurement* über die Win PDH lib.
- ⇒ Die Client Abschaltung werden auch für die ServerTools Initialisierungen angewendet:
HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\ERGOPlan PoolingServer\client_connect_timeout.
- ⇒ Das Beenden von ungenutzten Manufacturing Hub-Server Prozessen wird jetzt asynchron durchgeführt:▪ Dadurch hat sich die PoolingServer Konfigurationsdatei nicht geändert.
- Die ServerPools müssen nicht neu verbunden werden. Die ServerPools müssen nur dann neu verbunden werden, wenn die DCOM Verbindung oder die Rückmeldungen der load measurements nicht antworten. Diese Änderung reduziert effizient ein großes Anzahl von zu lange wartenden oder blockierten Client Anmeldungen.

ServerTools:

- ❑ Die Initialisierung eines Manufacturing Hub-Servers (inklusive Datenbankverbindung) erfolgt über die separate Initialisierungsmethode "IEP-Manufacturing Hub-ServerInstance::InitailizeManufacturing Hub-Server". Der ServerPool ruft diese Methode asynchron auf, um das Blockieren des ServerPooling vorzubeugen, wenn die Initialisierung des Manufacturing Hub Servers fehlschlägt, z. B.: wegen einer bevorstehenden, unvollständigen Datenbankverbindung (die Systemressourcen wurden reduziert und damit ist das Öffnungen von neuen Datenbankverbindungen behindert). Die Abschaltung für die Initialisierung vom Manufacturing Hub-Server kann mittels eines Registrierungseintrages konfiguriert werden:

HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\Ergoplan EPServer-Tools\serverpool\Manufacturing Hub-Server_creation_timeout = '120'.

Der Standardwert beträgt 2 Minuten (analog mit der Abschaltung der COM Instanzierung).

ServerTools-Client:

- ❑ Der Serverprozess 'UserExitPlugin.exe' wird für das Anzeigen und Beenden von PPR Server Prozessen verwendet.

DPE 5.R17

- ❑ Einführung von exklusiven Clientverbindungen:
 - ⇒ Wenn ein Client exklusiv sich verbindet, wird der Manufacturing Hub-Server nur für diesen Client reserviert.
 - ⇒ Um einen Server exklusiv zu verwenden, muss mit einem exklusiven Label gekennzeichnet sein, um die angeschlossene Benutzergruppe zu identifizieren.
Wenn das exklusive Label eindeutig ist, wird kein anderer Client mit dem exklusiv beschrifteten IPDServer Prozess verbunden.

PoolingServer:

- ❑ Die Konfiguration wird nur dann syntaktisch analysiert und interpretiert, wenn das Dateidatum sich geändert hat, - dies verkleinert die Prozessbelastung, da die Änderung der Konfiguration alle 30 Sekunden überprüft wird
- ❑ Die Fehlerprotokollierung beim Analysieren und Interpretieren der Konfigurationsdatei, ist durch Vermeiden von neuen Fehlerprotokolleinträgen für das Lesen einer unveränderten Konfiguration verbessert worden.
- ❑ Die Konfigurationssyntax wurde erweitert.
- ❑ Das automatische Priorisierung von Maschinen in der PoolingServer Konfiguration wurde geändert.

ServerTools-Client:

- Anzeigen der exklusiven Labels und des persistent Status.
format: <machine name> [PID: <process id>, LABEL: '<exklusiv label>', persistent]
Wenn persistent auf True gesetzt wird <persistent> wird es angezeigt, sonst nicht.
- ❑ Anzeige der Manufacturing Hub-Server Lock:
 - "out of memory" memory consumption has reached 2 GB minus the emergency failsafe reserved memory
 - "critical exception" a critical exception has occurred (e.g. Access-Violation)
 - "database exception" a critical database exception has occurred (e.g. database connection lost)
 - "RPC connection lost" PRC connection to updatemng, lockmng or security logger lost
 - "security log io" IO error while logging security data
 - "terminating" Manufacturing Hub-Server is being terminated by the Manufacturing Hub-Server termination routine of the server pool
 - "application" the Manufacturing Hub-Server is locked for further connections using the ServerPool API

Weiter Informationen werden in die Manufacturing Hub-Server error log Datei geschrieben.

DPE 5.17SP2**PoolingServer:**

- ❑ PoolingServer Konfiguration:

Optionen für Serverdomänen können auch auf Maschinenebene verwendet werden.

Mehrere Serverprozess- Notfallkonfigurationen können in der Konfigurationsdatei definiert werden. Identifiziert werden Sie über ihren konfigurierten Notfallnamen.

```
<<emergencyconfig>>
```

```
config_name = import_emergency
```