ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

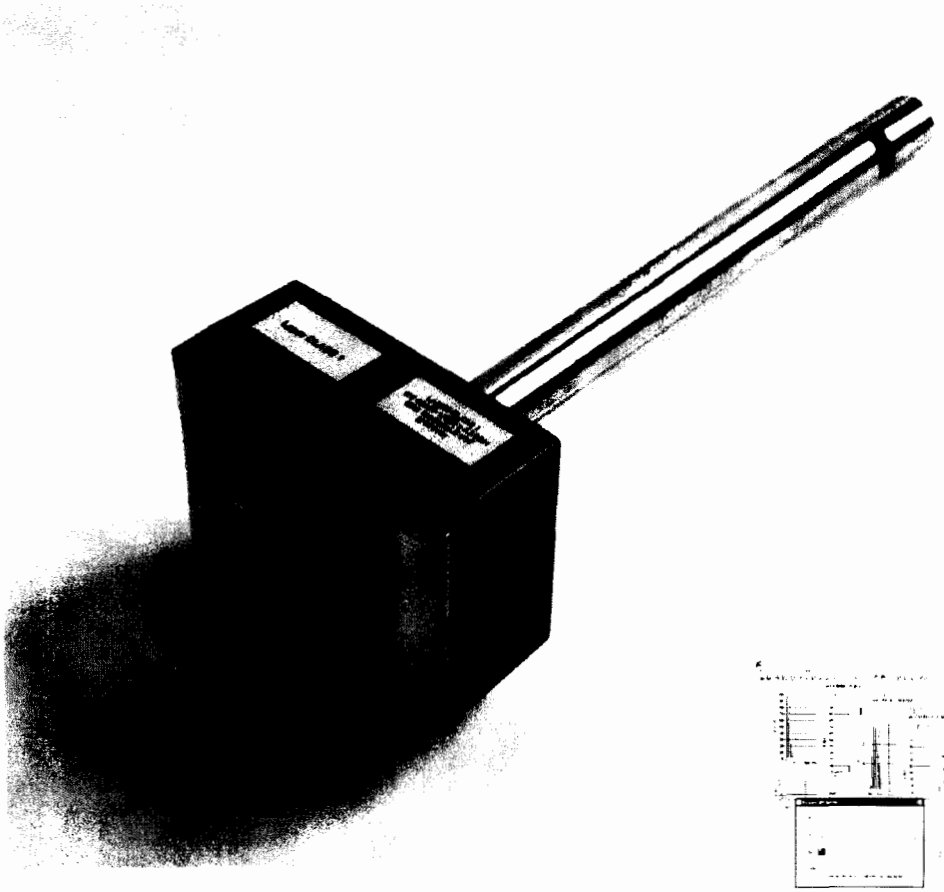
- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

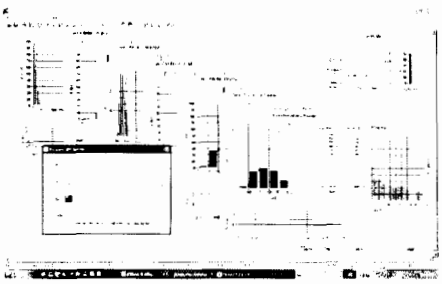
Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Malvern / Parsum IPP 70 VK1* at our website: *Click HERE*

parsum®
Gesellschaft für Partikel-, Strömungs- und Umweltmeßtechnik mbH





Manual

Parsum OPC Server for IPP 70

May 2008

parsum® - Gesellschaft für Partikel-, Strömungs- und Umweltmeßtechnik mbH
Reichenhainer Straße 34 – 36, 09126 Chemnitz, Germany
Tel. +49(0)371/26758690, Fax +49(0)371/26758699, Email: info@parsum.de, Internet: www.parsum.de

Contents

Page

1. Software licence regulation 2

2. General description 3

3. Installation of Parsum OPC server 3

4. Configuration of Parsum OPC server 4

5. Using the Parsum OPC server 6

6. Available tags 8

6.1. Reading: data, parameter and server status 8

6.2. Writing: measurement parameter 10

6.3. Writing: commands to control the measurement ... 11

7. Error codes 13

8. OPC client access – demo OPC client 14

9. Troubleshooting 16

10. FAQ 16

1. Software licence regulation

1. GRANTING OF A LICENCE.
   Parsum GmbH grants you a limited, non-exclusive right to use the software program with accompanying documentation purchased by you (the "SOFTWARE"). The validity of this regulation is linked solely to precisely those measurement probes which form part of the goods supplied to you. This Licence Agreement permits you to use a copy of the SOFTWARE on a single computer at your workplace. The SOFTWARE is "in use" on a computer if it has been loaded into temporary storage (i.e. RAM) or installed in permanent storage (e.g. on a hard disk, CD-ROM or other storage device). All rights not expressly granted to you in this Agreement are reserved to Parsum GmbH.

2. COPYRIGHT.
   The SOFTWARE is the property of Parsum GmbH and is protected by German copyright law and international agreements. You must therefore treat the SOFTWARE like all other copyright-protected documents. You may however either: (a) make a copy of the SOFTWARE solely for back-up or archiving purposes, or (b) transfer the SOFTWARE to a single hard disk if you are retaining the original solely for back-up or archiving purposes. The written material accompanying the SOFTWARE may not be reproduced without written consent from Parsum GmbH.

3. RESTRICTIONS.
   (A) Reverse engineering of the SOFTWARE is not permitted, nor is it permissible to decompile or disassemble the SOFTWARE.
   (B) Use of the SOFTWARE with probes other than precisely those forming part of the goods supplied to you is not permitted.
   (C) You may not hire the SOFTWARE out.

2. General description

The Parsum OPC server is an extension for the universal TCP server interface standardized included in the measurement software "Inline Particle Probe V7.12" or later version. It could be used with all Installations "Inline Particle Probe V7.12" or later.

Since the OPC server communicates with the measurement software via the TCP server interface, it is required, that the TCP server is activated in the measurement software "Inline Particle Probe V7.12" without Password Protection (for activation please look to the manual of the measurement software "Inline Particle Probe V7.12").

The OPC server could be installed at the measurement PC in addition beside the measurement software "Inline Particle Probe V7.12" or at an own PC in the network - the function is comparable.

3. Installation of Parsum OPC server

The installer installs the Parsum OPC server and its dependencies. The server is based on National Instruments LabVIEW™ and requires the following components, which are all included in the installer shipped with the OPC server.

- National Instruments Runtime Engine Version 8.5
- National Instruments Variable Engine 1.4
- National Instruments DataSocket 6.0

Note:
In addition the Parsum OPC server requires the Microsoft .NET 2.0 Framework installed. The framework is not a part of the OPC installer. If not already installed, you could find the separate setup for Microsoft .NET 2.0 in the ThirdParty folder of the install CD. The OPC server requires Microsoft Windows 2000 SP4 or later.

To install the OPC server please start "setup.exe" from the installation DVD and follow the instructions.

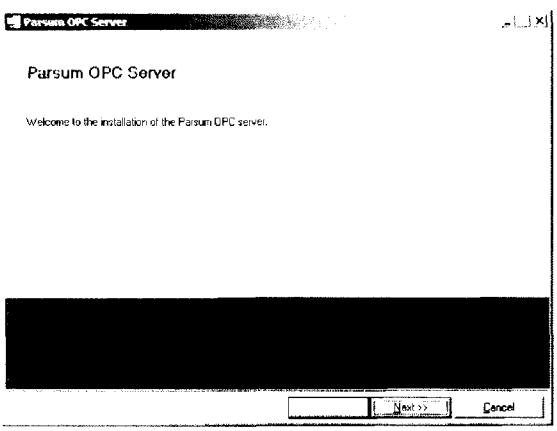


Figure 1: Installer

After the successful installation please restart the computer. Now you are ready to configure your probes described in the next chapter.

4. Configuration of Parsum OPC server

The Parsum OPC server could supply up to 24 single Inline Particle Probes IPP70. Thereby each probe is identified by the IP address of the connected measurement PC and by an individual port number adjusted for each single probe in the measurement software "Inline Particle Probe V7.12" or later.

The IP addresses and the port numbers for all connected probes are stored in two xml-files in the program folder of the OPC Server which are loaded at the start up of the Parsum OPC server. Changes in these xml-files during the runtime of the Parsum OPC server, have no effect to the existing connections.



At the first start of the Parsum OPC server after installation these xml-files contain mostly not valid default values. Therefore you should configure the xml-files for the connected probes prior the first start of the Parsum OPC server.

To configure your probes please start the also installed "Parsum OPC Configurator". You'll get a configuration dialog where you could set up all 24 probes individually and your desired update rate for the OPC tags. Depending on the number of active probes the time needed to update all probes may vary. Decreasing the update time will result in higher CPU usage by the Parsum OPC server. An update time between one and two seconds is recommended for most applications.

To set up a new probe please take the following steps:

➔ Start the "Parsum OPC configurator". A configuration dialog appears. If you call the configurator the first time you will see an empty configuration.

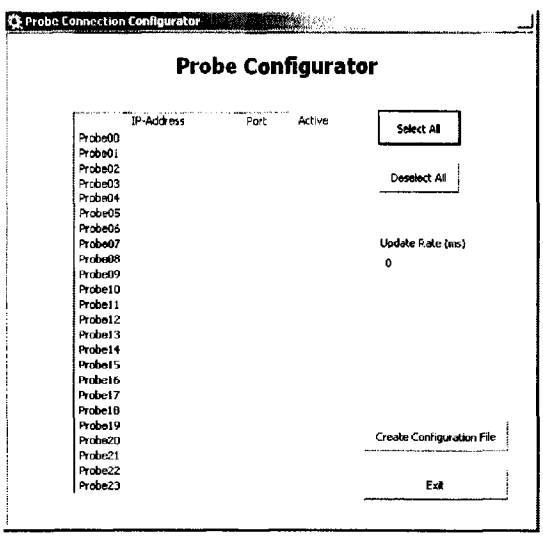


Figure 2: OPC Probe Configurator

➔ Now please enter following in one row for each connected probe:
(1) the **IP-Address** of the measurement PC connected with the Inline Particle Probe IPP70 as an IP-Number (e.g. 192.168.1.1) or a name (e.g. IPP-PC).
(2) the **Port number** of the individual probe adjusted in the measurement software Inline Particle Probe.
(3) a hook in the **Active** checkbox to include this probe in the update cycle of the Parsum OPC server
(4) the **Update Rate [ms]** for all OPC tags – all tags are updated with this time cycle (the required time varies by the number of activated probes and the speed of the Ethernet connection – usual range is 1000...2000ms)

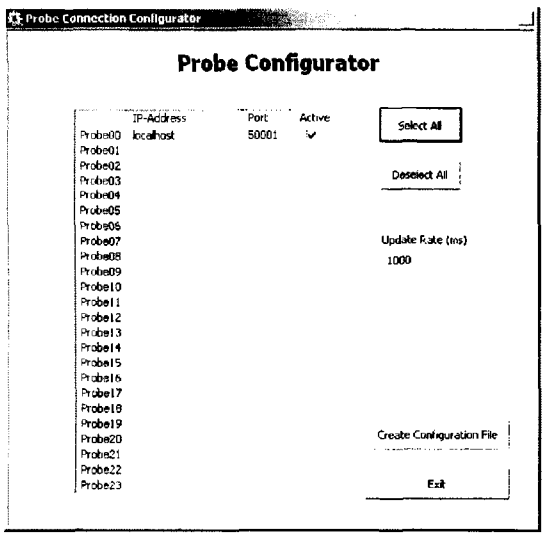


Figure 3: Configuring probe

➔ To create the xml-files please press the button *"Create Configuration File"*. To close the "Parsum OPC Configurator" without creating or renew the xml-files (e.g. if you wanted to inspect the settings only), please press the button *"Exit"* without *"Create Configuration File"*.
The progress bar indicates the current progress of creating/writing the xml-files and the needed time will vary by the number of activated probes.

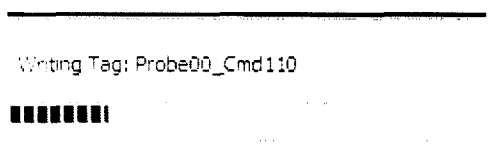


Figure 4: Progress

After successful creation of the xml-files please leave the "Parsum OPC Configurator" with the button "Exit". Now the xml-files are ready to use for the Parsum OPC server.

5. Using the Parsum OPC server

When you start the OPC server a splash screen (see Figure 5: Splash Screen) will appear while configuring the internal data of the server. You will see a progress bar to show the current progress. The time needed for this operation may also vary depending on the number of activated probes.



Figure 5: Splash Screen

Along this activation the OPC server creates all single connections to the activated probes in the network, i.e. to the activated TCP server at the measurement PCs.

A successful created connection to a probe (i.e. to its activated TCP server) could be easily checked in the taskbar of the Mess-Panel of this probe in the measurement software "Inline Particle Probe V7.12". The right textbox normally shows "No client connection" if the TCP server is not connected to and external client and e.g. "Connected with 192.172.1.3" if the connection is successful created (e.g. to the Parsum OPC server).

If a probe is not reachable at the start or during continuously operation (e.g. a measurement PC is switched off or have to be reboot during operation) the Parsum OPC server tries to reconnect to this specific probe once a minute. Thereby the connection status is indicated by the OPC tag "GetConnectionStatus" (see chapter 4).

After starting the Parsum OPC server the main screen of the running server appears (see Figure 6: Parsum OPC server).

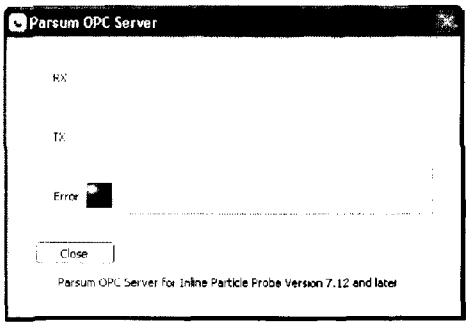


Figure 6: Parsum OPC server

This main screen has 3 LED to indicate the current status of the Parsum OPC server:

- ➔ The RX & TX LED indicates TCP/IP traffic from & to the connected TCP server in the measurement software (i.e. from & to the activated probes).
- ➔ The error LED will go to red, if an error was thrown by the OPC server. This may indicate TCP/IP connection problems (see chapter 7 – Error codes).

Press the *Close* button to close the OPC server and free the connection to the TCP server of the measurement software.

6. Available tags

The mainly available tags (read/write) are corresponding to the functionality of the TCP server version 1.22 in the measurement software "Inline Particle Probe V7.12" (Get/Set/Cmd).

6.1. Reading: data, parameter and server status

These tags supply all results from the measurement software "Inline Particle Probe V7.12", the currently adjusted measurement settings and additional status information from the OPC server.

Level and generally		
GetVersion	version of TCP-server in the IPP-software	
GetCh0Dyn	channel 0 dynamic of the probe signal	**[%]**
GetCh1Dyn	channel 1 dynamic of the probe signal	**[%]**
GetLoading	particle loading in the measurement volume	**[%]**
GetPartRate	particle rate (properly found particles)	**[Part/s]**
GetPulsRate	pulse rate (all particles in the signal)	**[Puls/s]**
GetPartAnz	current particle count in the ring buffer	**[Part]**
GetNoPart	Flag → "No particle"	
GetTpIppCbCmPerH	throughput in the measurement volume	**[cm^3/h]**
Sieve distribution		
GetFrac00...10	fraction of one adjusted sieve (00...10)	**[%]**
GetFracAll	list with all fractions (ARRAY)	**[%]**
GetPass00...10	passage of one adjusted sieve (00...10)	**[%]**
GetPassAll	list with all passages (ARRAY)	**[%]**
Static particle properties		
GetX01_0 (...X10_0 ff)	char. property of the Q0 size-distribution	**[µm]**
GetX01_3 (...X10_3 ff)	char. property of the Q3 size-distribution	**[µm]**
GetX01_Velo (...X10_Velo ff)	char. property of the Q0 velocity-distribution	**[mm/s]**

NOTE:
Available for all GetX-Commands are "X01, X10, X25, X50, X63, X75, X90 and X99"

Variable particle properties (quantils)		
GetQuantil_0	adjusted quantil in Q0-size distribution	**[%]**
GetQuantil_3	adjusted quantil in Q3-size distribution	**[%]**
GetQuantil_Velo	adjusted quantil in Q0-velocity distribution	**[%]**
GetXQuantil_0	characteristically property to „Quantil_0"	**[µm]**
GetXQuantil_3	characteristically property to „Quantil_3"	**[µm]**
GetXQuantil_Velo	characteristically property to „Quantil_Velo"	**[mm/s]**
Properties of LIN-Size distribution		
GetLinSizeKlassBrt	width of all linear size classes	**[µm]**
GetLinSizeKlassAnz	count of all linear size classes	
GetLinSizeKlassOg	upper ends of all linear size classes (ARRAY)	**[µm]**
GetLinSizeKlassMt	middles of all linear size classes (ARRAY)	**[µm]**
Q0 (size) distribution (LIN)		
GetLinQ0SizeSumme	linear Q0 (sum) size-distribution (ARRAY)	**[%]**
GetLinQ0SizeDichte	linear q0 (density) size-distribution (ARRAY)	

Q3 (size) distribution (LIN)

GetLinQ3SizeSumme	linear Q3 (sum) size-distribution (ARRAY)	**[%]**
GetLinQ3SizeDichte	linear q3 (density) size-distribution (ARRAY)	

Properties of the LOG-Size distribution

GetLogSizeKlassBrt	width of all logarithmic classes as „Δ log(x)"	
GetLogSizeKlassAnz	count of all logarithmic classes	
GetLogSizeKlassOg	upper ends of all log. size classes (ARRAY)	**[µm]**
GetLogSizeKlassMt	middles of all log. size classes (ARRAY)	**[µm]**

NOTE:
The resulting "GetLogSizeKlassOg/Mt"-arrays include (logarithmic distributed) Size-Values **as [µm]** to use it for the X-Axis of a distribution graphic. It does **not** includes the size **as [lg(size)]**. Anyway the distance between the array-elements is a constant logarithmic classwidth „Δ log(x)" available in the OPC tag *"GetLogSizeKlassBrt"*.

Q0 (size) distribution (LOG)

GetLogQ0SizeSumme	logarithm. Q0 (sum) size-distribution (ARRAY)	**[%]**
GetLogQ0SizeDichte	logarithm. q0 (density) size-distribution (ARRAY)	

Q3 (size) distribution (LOG)

GetLogQ3SizeSumme	logarithm. Q3 (sum) size-distribution (ARRAY)	**[%]**
GetLogQ3SizeDichte	logarithm. q3 (density) size-distribution (ARRAY)	

Properties of the LIN-Velo distribution

GetLinVeloKlassBr	width of all linear velocity classes	**[mm/s]**
GetLinVeloKlassAnz	count of all linear velocity classes	
GetLinVeloKlassOg	upper ends of all linear velo. classes (ARRAY)	**[mm/s]**
GetLinVeloKlassMt	middles of all linear velocity classes (ARRAY)	**[mm/s]**

Q0 (velo) distribution (LIN)

GetLinVeloSumme	linear Q0 (sum) velocity-distribution (ARRAY)	**[%]**
GetLinVeloDichte	linear q0 (density) velocity-distribution (ARRAY)	

Simulation parameter

GetSimEnabled	Flag → "Simulation-Mode"
GetSimShowAbortPanel	Flag → „display of the Sim-Abort Panel"

Measurement parameters

GetMessStarted	Flag → "measurement started"

In addition to the measurement results and current measurement settings the Parsum OPC server provides also two tags with status information from the OPC server:

GetLastError error number according to the error codes in chapter 7
This tag indicates the success resp. the fail status of the last Set or Cmd operation (see chapter 4.2 and 4.3). If this tag is zero the operation succeeded. Otherwise GetLastError provides the last error code returned by the TCP server of this probe in the measurement software (see chapter 5 – error codes).

GetConnectionStatus Flag -> "connection to the TCP server established"
This tag indicates the status of the connection to the TCP server in the measurement software for this single probe. If this tag is TRUE, the connection is established OK. If the

tag is FALSE, the connection is not yet established. In this case, the OPC server tries to reconnect continuously once in a minute until the OPC server will be closed by the user.



It is recommended to check the **GetLastError** tag after all Set and Cmd operations to be sure, that the operation was successfully executed.



It is recommended to check the **GetConnectionStatus** tag before all accesses to the OPC server (Set/Get/Cmd). If the GetConnectionStatus tag is FALSE all follow operations and available data will be invalid.

6.2. Writing: measurement parameter

The following tags are used to configure the measurement software for this single probe by setting of several measurement parameters. Changes on tags are immediately sent to the measurement software independent of the adjusted Update Rate [ms] of the OPC server (see chapter 2).

SetLinSizeKlassBrt	size class width	**[µm]**
SetLinVeloKlassBrt	velocity class width	**[mm/s]**
SetSizeUg	lower limit of size range	**[µm]**
SetSizeOg	upper limit of size range	**[µm]**
SetVeloUg	lower limit of velocity range	**[mm/s]**
SetVeloOg	upper limit of velocity range	**[mm/s]**
SetAdjustFactor	adjust factor (calibration)	
SetAdjustOffset	adjust offset (calibration)	**[µm]**
SetRingLaenge	particle buffer length (elements)	**[particles]**
SetMaxLoading	maximal loading	**[%]**
SetSampleRate	sample rate of the DAQ	**[Hz]**
SetPos	Flag → „activation of ProbOfStay"	
SetCoincidenceLevel	coincidence level	**[%]**
SetQuantil_0	Quantil of Q0-size distribution	**[%]**
SetQuantil_3	Quantil of Q3-size distribution	**[%]**
SetQuantil_Velo	Quantil of Q0-velocity distribution	**[%]**
SetSimEnabled	Flag → „activation of Simulation-Mode"	
SetSimShowAbortPanel	Flag → „display of the Sim-Abort Panel"	

SetSieveList <0> <Sieve1> ... <Sieve10>
complete list of all 11 sieves (**0 µm** ... maximum) (ARRAY) **[µm]**



To determine the success of your desired operation it is recommended to read the **GetLastError** tag. If this tag is zero the operation succeeded. Otherwise GetLastError provides the last error code returned by the TCP server of this probe in the measurement software (see chapter 7 – error codes). The changes to this tag are reverted in this case.

The OPC server gets all measurement parameters for the corresponding tags from the measurement software during of the start up sequence while the TCP connection is established. Thereafter the OPC server updates the parameter-tags according to successful changes in these tags (transmitted to the measurement software). The OPC server does not reread the parameters from the measurement software during its runtime. Therefore it is important to know, that all parameter-changes manually made in the user interface of the measurement software, will not be seen in the current parameter-tags of the OPC server. Manually changes will be reread in this case during a restart (or a reconnection) of the OPC server.

Therefore please be sure to make no manually changes in the measurement software, if the measurement is under the control of the OPC server resp. under the control of the OPC client!

6.3. Writing: commands to control the measurement

The following tags are used to control operation of the probe (the measurement).

Generally there are two kinds of possible commands in the OPC server:
➔ "one-tag-commands" **without** a parameter (e.g. CmdStart_exec; CmdStop_exec ...)
➔ "two-tag-commands" **with** a parameter (e.g. CmdSaveDa1File_exec ...)

Thereby the necessary parameter for a "two-tag-command" is provided in a corresponding second tag (e.g. CmdSaveDa1File; CmdSaveDa1File etc.). The corresponding parameter-tag has the same name like the command-tag without the ending "_exec". Please be sure, to set the corresponding parameter-tag to the required value **before execute** the two-tag-command with the ending "_exec".

In all cases a command will be executed, if the corresponding "*_exec" tag is changed from FALSE to TRUE. A change from TRUE to FALSE has **no** effect. After execution the OPC server doesn't resets the "*_exec" tags to FALSE autonomous. It is the responsibility of the OPC client (i.e. the user) to reset the "*_exec" tag to FALSE to prepare the tag for the next execution.

"one-tag-commands"	
CmdStart_exec	starts the measurement
CmdStop_exec	stops the measurement
CmdHide_exec	hides the program GUIs
CmdDisplay_exec	displays the program GUI
CmdLoadDefaultSetting_exec	loads/activates „Default-Parameter" for Setting-Area
CmdLoadDefaultService_exec	loads/activates „Default-Parameter" for Service-Area
CmdLoadDefaultAll_exec	loads/activates „Default-Parameter" for Setting and Service-Area
CmdExit_exec	stops the measurement, saves all parameter and closes the complete measurement software "Inline Particle Probe V7.12"

"two-tag-commands"	
CmdSaveDa1File_exec	save the current measurement as „da1-File"
CmdSaveDa1File	parameter: filename for saving (*.da1)
CmdSaveDa2File_exec	save the current measurement as „da2-File"
CmdSaveDa2File	parameter: filename for saving (*.da2)

CmdLoadCfgFile_exec	load and activate parameter from a saved „cfg-File"
CmdLoadCfgFile	parameter: filename for loading (*.cfg)
CmdSaveCfgFile_exec	save the current set of parameters in a „cfg-File"
CmdSaveCfgFile	parameter: filename for saving (*.cfg)
CmdAcqSimFile_40_exec	acquire/save simulation data with 40 Waveforms
CmdAcqSimFile_40	parameter: filename for saving (*.sim)
CmdAcqSimFile_100_exec	acquire/save simulation data with 100 waveforms
CmdAcqSimFile_100	parameter: filename for saving (*.sim)
CmdAcqSimFile_500_exec	acquire/save simulation data with 500 waveforms
CmdAcqSimFile_500	parameter: filename for saving (*.sim)
CmdAcqSimFile_1000_exec	acquire/save simulation data with 1000 waveforms
CmdAcqSimFile_1000	parameter: filename for saving (*.sim)
CmdAcqSimFile_5000_exec	acquire/save simulation data with 5000 waveforms
CmdAcqSimFile_5000	parameter: filename for saving (*.sim)
CmdRelSimFile_exec	load of simulation data from a saved simulation
CmdRelSimFile	parameter: filename for reload (*.sim)

To determine the success of the desired operation it is recommended, to read the **GetLastError** tag. If this tag is zero the operation succeeded. Otherwise GetLastError provides the last error code returned by the TCP server of this probe in the measurement software (see chapter 7 – Error codes) and the command was not executed.

The following example illustrates the behaviour of executing commands to save the configuration in a file two times in two different files Test1.cfg and Test2.cfg.

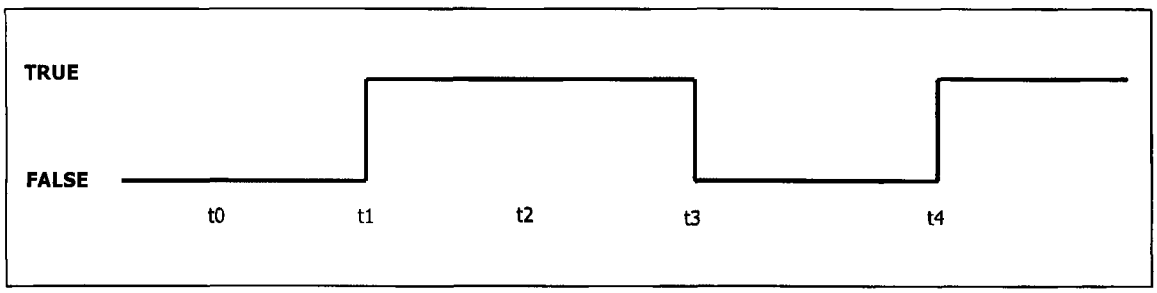


Figure 7: Executing commands behaviour

In this example we want to save the current set of parameters two times in different files "Test1.cfg" and "Test2.cfg".

t0: Set tag ***CmdSaveCfgFile*** to C:\\Test1.cfg
t1: Set tag ***CmdSaveCfgFile_exec*** from FALSE to TRUE
→ the parameters are saved in C:\Test1.cfg

t2: Set tag ***CmdSaveCfgFile*** to *C:\\Test2.cfg*
t3: Reset tag ***CmdSaveCfgFile_exec*** from TRUE to FALSE
t4: Set tag ***CmdSaveCfgFile_exec*** from FALSE to TRUE
→ the parameters are saved in C:\Test2.cfg

7. Error codes

This chapter lists all error codes that could be returned from the from the the probe i.e. measurement software.

ErrorCode	Description
0	No Error
5001	connection already active
5002	request password
5003	wrong password
5004	no command received (empty string)
5005	missing parameter
5006	unknown command
6000	unknown error
5020	active measurement - set parameter not possible
5021	Warning: KlassBrt (class width) set to "NULL" - default value loaded
5022	Warning: KlassBrt (class width) to large - KlassBrt narrowed
5023	Warning: KlassBrt (class width) to small - Range narrowed
5024	quantil out of range (0% ... 100%)
5030	Start - no DAQ hardware found (measurement with particle probe)
5031	Start - no simulation data loaded (simulated measurement)
5040	filename is empty
5050	active measurement - loading default parameter not possible
5051	cfg-file corrupted or not compatible (former parameters reloaded)
5052	cfg-file corrupted and reload of former parameters failed - default parameters loaded
5060	storage failed - disk full, no access, ...
5070	active measurement - loading of default parameter not possible
5080	simulation mode is active - acquire/save sim-data not possible
5081	not enough memory (too much waveforms)
5082	no measurement card found
5083	data acquisition manually aborted
5084	creating sim-file failed
5085	data storage manually aborted
5090	active measurement - reload of sim-data not possible
5091	open sim-file failed (not existing, file attributes, ...)
5092	not enough memory for this sim-file
5093	reloading sim-data manually aborted
5094	CRC-error, sim-file corrupted (incompatible, ...)

8. OPC client access – demo OPC client

The access via OPC is demonstrated using the OPC demo client from Softing. You can find the demo at the ThirdParty folder of your installer or you can download it at http://www.softing.com.

When you start the OPC Client first choose the Parsum OPC Server. It is located under DataAccess V2 or DataAccess V3. Choose National Instruments Variable Engine OPC Server → National Instruments.Varibale Engine.1 .

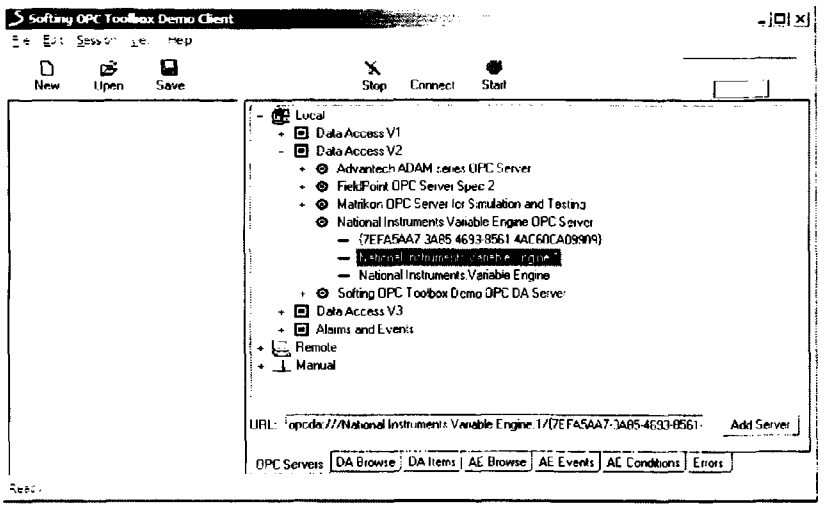


Figure 8: OPC client – Choose Server

Double-Click the selected server. It will appear in the left box. Switch to the DA Browse tab and unfold the server. You will see all your configured probes in the format ProbeXX (see Figure 9: OPC client - Choose Probe)

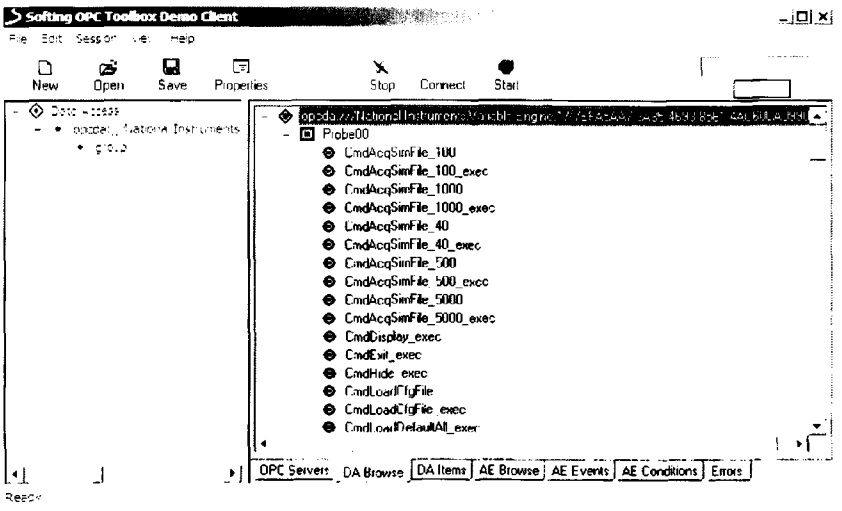


Figure 9: OPC client - Choose Probe

Double-clicking on the desired values will set up the tags in the left box. Switch to the DA Items tab. You will see the actual values of the probe.

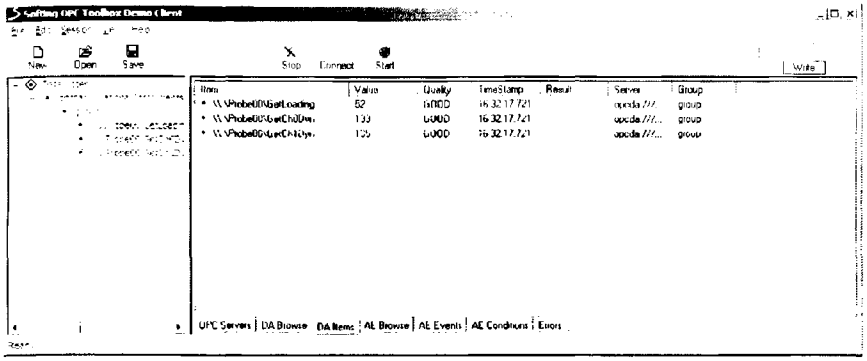


Figure 10: OPC client – Choose Tag

To write a new setpoint to the probe (e.g. sample rate) simply write a new value to the corresponding Set Tag.

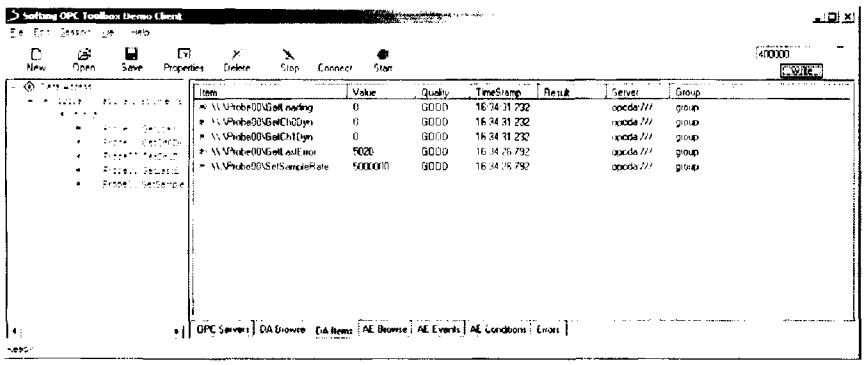


Figure 11: OPC client - Writing new setpoint

9. Troubleshooting

The OPC server writes an error log file for each day. It is located in the subdirectory LOG. This helps troubleshooting the OPC server.

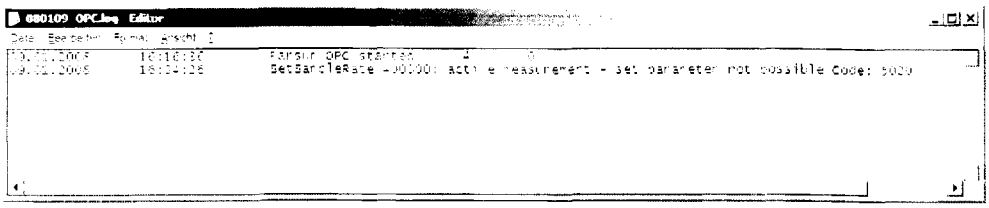


Figure 12: Sample Log File

10. FAQ

Q: I configured my probes but I don't see actual values of the probe.
A:

1. Verify that the probe is accessible via TCP/IP.
2. Verify that the OPC server is running.
3. Verify the GetConnectionStatus variable is true.
4. Verify the measurement is started.
5. Review the OPC server's log file.

Q: I want to use the OPC server in a workgroup environment. Are there any considerations?
A:

1. The OPC server service and the OPC client service must run under the same account (username and password). The OPC service is the National Instruments Variable Engine.

Q: Are there any other considerations for my DCOM settings.
A:

1. Review the attached document *Using OPC via DCOM with XP SP2 v1.10.pdf.*

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP