

# **DataServer**

## **Version 1.05**

### **User Guide**

**Nanometrics Inc.**  
**Kanata, Ontario**  
**Canada**

© 2001–2004 Nanometrics Inc. All Rights Reserved.

## DataServer Version 1.05 User Guide

The information in this document has been carefully reviewed and is believed to be reliable for Version 1.05.xx. Nanometrics, Inc. reserves the right to make changes at any time without notice to improve the reliability and function of the product.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Nanometrics Inc.

This product uses the Java™ 2 Runtime Environment (J2RE) Standard Edition, which includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>

Nanometrics, Inc.  
250 Herzberg Road  
Kanata, Ontario, Canada K2K 2A1  
Tel (613)592-6776  
Fax (613)592-5929  
Email [info@nanometrics.ca](mailto:info@nanometrics.ca)

**Part number** 14835R3

**Release date** 2004-09-13

# Contents

1	Installing DataServer	1
1.1	Requirements	1
1.2	Install DataServer	2
2	Running DataServer	2
2.1	Setting DataServer to start automatically	2
2.2	Starting and stopping DataServer manually	3
2.3	Using the DataServer run-time commands	3
2.4	Monitoring DataServer operation	3
3	Configuring DataServer	4
3.1	Configure DataServer	4
3.2	Definition of configuration file parameters	4
3.2.1	[ Log ]	4
3.2.2	[ TcpServer ]	5
3.2.3	[ EventData ]	5
3.2.4	[ Users ]	6
3.3	Example DataServer configuration file	6



---

# DataServer

---

DataServer is a utility program that provides local and remote access to Nanometrics seismic, serial, and state-of-health data via TCP/IP. It accepts TCP connections from client programs and forwards data in response to specific requests. DataServer provides access to data stored in ringbuffers or other files, and thus complements the realtime data service provided by NaqsServer online Datastreams.

DataServer reads the Naqs station file `Naqs.stn` to determine name and location of data files to be accessed. Data are forwarded in the original compressed packet format received from the acquisition equipment. All data requests and responses follow the Nanometrics Data Access Protocol.

DataServer provides client programs with these types of data:

- ♦ Time-series data from the NaqsServer ringbuffers
- ♦ State of Health (SOH) from the NaqsServer ringbuffers
- ♦ Serial data from the NaqsServer ringbuffers
- ♦ Naqs triggers from the Naqs event files (if enabled in Naqs)
- ♦ Naqs events from the Naqs event files
- ♦ Time-series, SOH, and serial data from a source other than Naqs (for example, Orion)

## 1 Installing DataServer

DataServer must be installed either on the NaqsServer machine, or on a machine that has TCP/IP access to the Naqs machine and network access to the data files (Naqs ringbuffers and event files). To simplify the configuration, install and run DataServer in the same working directory as NaqsServer.

Once you have installed DataServer, check the parameter settings in the DataServer configuration file and the applicable settings in the `Naqs.stn` file. Edit these, if necessary, to conform to your network (see Section 3).

### 1.1 Requirements

- ♦ Windows or Solaris operating system
- ♦ Java Runtime Environment 1.4.2\_0x or higher
- ♦ Patches as may be recommended for the OS and JRE

- ◆ Nanometrics DLLs/Libraries 1.85 or higher
- ◆ Station file for Naqs 1.80 or higher
- ◆ 256 MB RAM (minimum)
- ◆ Minimum hard disk space (these requirements do not include space required for the operating system):
  - Nanometrics application code: 5 MB
  - Log file capacity: allow 5 MB (with typical use). Normally, log files older than 30 days may be deleted.
  - JRE: 50MB

## 1.2 Install DataServer

On Windows:

1. From either a command prompt or Windows Explorer, open the installation CD directory `Win32\DataServer\version number`
2. Copy all files from the `bin` directory into the `c:\nmx\bin` directory, and all files from the `user` directory into the `c:\nmx\user` directory (these directories are created on your computer during DLL installation).
3. Check the system path, and add the directory `c:\nmx\bin` if it is not already included.
4. Adjust the configuration parameters as outlined in Section 3.

On Solaris and Linux:

- ▶ See the installation instructions for the acquisition system workstation.

## 2 Running DataServer

In a typical network, DataServer will be set up to start automatically using the Nmx-Watchdog program (on Windows) or scripts (on Solaris and Linux). It can also be started and stopped manually from the command line. Once DataServer is running, you can use the run-time commands to change the log message verbosity, to create a new log file, and to stop DataServer.

### 2.1 Setting DataServer to start automatically

On Windows:

- ▶ Check the `watchdog.ini` file to see whether a section exists to have DataServer started and monitored automatically by NmxWatchdog. If it is not in the file, add a section:

```
[ WatchEntry n ]
ProgramTitle = DataServer
ProgramPathname = "java -cp c:\nmx\bin\DataServer.jar DataServer"
WorkingDirectory = "c:\nmx\user"
ExitAction = Restart
PingsSemaphore = true
StartDelay = 6s
```

On Solaris and Linux:

- ▶ See the information on general utilities for the acquisition system workstation.

## 2.2 Starting and stopping DataServer manually

It is important that DataServer be shut down properly in order for it to release its system resources.

- ▶ To start DataServer, enter `dataserver` in any terminal window.
- ▶ To stop DataServer, enter `quit` in the DataServer terminal window.

## 2.3 Using the DataServer run-time commands

DataServer supports a basic keyboard interface for entering run-time commands, with the command options described in Table 2-1.

- ▶ Enter run-time commands in the DataServer terminal window: Enter *command*

**Table 2-1** DataServer command options

To do this...	Enter this command...
Display all log messages in the log file; set the log verbosity to DEBUG	<code>d</code>
Suppress debug messages in the log file; set the log verbosity to VERBOSE	<code>v</code>
Suppress debug and verbose messages in the log file; set the log verbosity to INFO	<code>i</code>
Move the log file (close the current log and start a new file)	<code>m</code>
Stop DataServer and exit: ###on Windows on Solaris	<code>quit</code> <code>stop</code>

## 2.4 Monitoring DataServer operation

DataServer generates log messages that trace the operation of the program. It displays these messages in the terminal window, and writes them to the DataServer log file. You can set the level of detail (the verbosity) of the information to be displayed and recorded.

- ▶ To view the log, open the log file *LogFilename\_date.log* (for example, *DataServer\_20030331.log*) in a text editor. The log file name and location are set in the [ Log ] section of the DataServer configuration file.
- ▶ To set the verbosity of log messages on startup, edit the [ Log ] section of the DataServer configuration file.
- ▶ To change the verbosity of log messages while DataServer is running, use the run-time commands.

## 3 Configuring DataServer

DataServer reads the `Naqs.stn` file in its working directory to determine the name and location of ringbuffer files to be accessed. If DataServer is run from the same machine and working directory as NaqsServer, then it will be able to access automatically all of the data recorded by NaqsServer.

You can use DataServer to access data from a source other than Naqs (for example, Orion), by creating a `Naqs.stn` file to provide the names and locations of the ringbuffers for that data source.

### 3.1 Configure DataServer

1. Ensure DataServer has access to the required files:
  - a) Access to configuration files (`Naqs.stn`, `dataserver.ini`): This is automatic if DataServer is run in the same directory as Naqs.  
If you are using DataServer to access data from a source other than Naqs (for example, Orion):
    - ▶ Create a `Naqs.stn` file in the same directory as DataServer, and then edit the `Naqs.stn` file to provide the names and locations of the ringbuffers for the data source.
  - b) Network access to data files (ringbuffers and event files): These file locations are defined in the `Naqs.stn` file.
2. Set external parameters as required:
  - ▶ If you wish to make trigger data available, in the `Naqs.stn` file set the parameter *SaveTriggers* to *Yes*.
3. Set DataServer parameter values in the DataServer configuration file (`dataserver.ini`) as described in Section 3.2.

### 3.2 Definition of configuration file parameters

The `dataserver.ini` file contains these sections:

- [ Log ]
- [ TcpServer ]
- [ EventData ]
- [ Users ]

All sections and parameters are required. There must be exactly one section of each type. All sections, and all parameters within sections, must be in the same order as listed below. The format for each entry is *Parameter = Value*. For example, `Directory = events`. See also the example configuration file in Section 3.3.

#### 3.2.1 [ Log ]

The [ Log ] section defines the location, name, and verbosity of the DataServer log file. It contains the parameters described in Table 3-1.



**Table 3-1** [ Log ] section parameters

Parameter	Definition
<i>LogFilename</i>	The name of the DataServer log file. A date-stamped log file based on this name is created each day. The file name uses the format <i>baseName_yyyymmdd.ext</i> where <i>baseName</i> and <i>ext</i> are the name and extension of the filename specified. For example, <code>DataServer_20030508.log</code> <ul style="list-style-type: none"> <li>Permitted values: Any valid file name.</li> </ul> Example: <code>DataServer.log</code>
<i>LogDirectory</i>	The pathname for the directory in which to store the log file. <ul style="list-style-type: none"> <li>Permitted values: Any valid pathname. Do <i>not</i> include the trailing slash.</li> </ul>
<i>Verbosity</i>	The startup verbosity of the log file. <ul style="list-style-type: none"> <li>Options are DEBUG, VERBOSE, INFO</li> </ul>

### 3.2.2 [ TcpServer ]

The [ TcpServer ] section defines the configuration of the TCP server socket that client programs will use to request data. It contains the parameters described in Table 3-2.

**Table 3-2** [ TcpServer ] section parameters

Parameter	Definition
<i>Port</i>	The TCP port on which DataServer listens for client connections. This can be configured to avoid conflicts with other services, and to meet firewall requirements. <ul style="list-style-type: none"> <li>Permitted values: Any available port number.</li> </ul>
<i>MaxConnections</i>	The maximum number of concurrent connections to support. Most client programs connect for only a few seconds at a time, so setting this value to a small number of concurrent connections should suffice. <ul style="list-style-type: none"> <li>Permitted values: Any integer from 1 to 10.</li> </ul>

### 3.2.3 [ EventData ]

The [ EventData ] section defines the directory and file names of the event and trigger data that is to be read. It contains the parameters described in Table 3-3.

**Table 3-3** [ EventData ] section parameters

Parameter	Description
<i>Directory</i>	The pathname for the directory in which event files are stored. This is treated as a relative pathname—that is, relative to the directory in which NaqsServer is running—unless it is specified as an absolute pathname (with a leading slash).  This should normally be set to the same value as the EventPath parameter in the <code>Naqs.ini</code> file. <ul style="list-style-type: none"> <li>Permitted values: Any valid pathname. Do <i>not</i> include the trailing slash.</li> </ul>

**Table 3-3** [ EventData ] section parameters (Continued)

Parameter	Description
<i>Filename</i>	<p>The base name of the NaqsServer event files. DataServer looks for date-stamped event files named <i>baseName_yyyymmdd.ext</i> where <i>baseName</i> and <i>ext</i> are the name and extension of the filename specified. For example, <i>Naqs_20030328.elf</i></p> <ul style="list-style-type: none"> <li>Permitted values: Any valid file name, but should normally be the same as the <i>EventFile</i> parameter in the <i>Naqs.ini</i> file; typically, <i>Naqs.elf</i>.</li> </ul>

### 3.2.4 [ Users ]

The [ Users ] section defines user names and passwords for users authorized to access this data service. Client programs connecting to DataServer must provide a valid user-name and password. This section can contain any number of entries of the form describes in Table 3-4.

**Table 3-4** [ Users ] section parameter

Parameter	Description
<i>username</i>	<p>The name of an authorized user (up to 11 characters). Set <i>username = password</i>, where <i>password</i> is the password for the specified user. Both username and password are case-sensitive.</p> <ul style="list-style-type: none"> <li>To allow unrestricted access, specify username <i>any</i> with password <i>none</i>.</li> </ul>

## 3.3 Example DataServer configuration file

This section contains an example DataServer configuration file (*dataserver.ini*). You can edit the *dataserver.ini* file in any text editor.

The inifile reader ignores white space and blank lines, so white space can be added anywhere within the file if desired to improve readability. Also, the inifile reader recognizes the double-slash “//” as a comment delimiter. You can use this to add comments anywhere in the file, and to temporarily remove parameters or sections from the file.

```
// DataServer.ini
// Config file for the DataServer program v1.00
// Modified June 22, 2001

[ Log ]

LogFilename = DataServer.log
LogDirectory = logs
Verbosity = Verbose

[ TcpServer ]

Port = 28002
MaxConnections = 4

[ EventData ]

Directory = data
```

```
Filename = Naqs.elf
```

```
[ Users ]
```

```
joe = passwd
```

```
bill = passwd2
```

```
//any = none
```

