# DNP Test Set User Guide

Revision Table

Change these bookmark definitions each time the document is revised and a Revision Entry is made in the table below. See the **INSTRUCTIONS ON USING THIS DOCUMENT AS A TEMPLATE** section below for instructions on how to change the bookmark definition.

Latest\_Rev: n/a

Rev\_Date: n/a

Entries in the revision table should be formatted with the “Revision Entry” or “Revision Entry Text” styles.

|  |  |  |
| --- | --- | --- |
| **Author** | **Date** | **Comments** |
| Matt Homovich | 06/16/10 | Initial Issue |
| Adam Crain | 06/16/10 | Q/A before committing to source control |
|  |  |  |

The DNP Test Set (TS) is a powerful tool for verifying DNP communications between masters and outstations. The DNP TS installation includes default settings for master and slave configurations. Custom configurations are performed by editing or replicating the default configuration files. The TS supports communication paths via serial or TCP/IP.

# Table of Contents

1. Requirements
2. [Installation](#_Installation)
3. Using [The](#_Using_the_Command) Command Line Arguments
4. [Xml Configuration File](#_xml_Configuration_File)s
5. [Executing the TS Master and Slave](#_Executing_the_TS)
6. [Test Set Operation](#_Test_Set_Operation_1)

# Requirements

# This user documentation focuses on the Windows installation. The core code base can be compiled for additional platforms. For more information, visit the project website:

# <http://code.google.com/p/dnp3>

# Installation

The TS installation package includes the installation of the master and slave test set modules as well as the Microsoft C++ Libraries required to run the TS modules. To install the TS, execute the Setup.exe file. The installer will guide the user through the installation process. If it is not already present on the machine, the installer will first install the *Microsoft Visual C++ 2005 SP1 Redistributable Package.* The installer will then load the test set and create both a master and slave shortcut in the Windows Start Menu. If the redistributable package does not install automatically, run vcredist\_x86\vcredist\_x86.exe manually.

The TS is a versatile application that uses an xml configuration file for its parameters. There is an xml configuration file for both the master and slave. The installer does not create the xml configuration files, default files are created for the master and slave when the master and slave modules are executed for the first time from the Windows Start Menu. The default location for the TS executable and configuration files is *<Program Files>\dnp3.*

# Using the Command Line Arguments

In addition to the Windows Start Menu shortcuts created by the installer, the TS can be executed and manipulated from the command line interface. Below is the complete list of command line arguments and their function, including some examples.

|  |
| --- |
| ***-H***  Help argument that displays the TS command line argument options.  *Example: C:\Program Files\dnp3\Testset.exe –H* |
| ***-F <filename>***  Specifies the xml configuration file to use. This function allows you to select which configuration file theTS will use. It allows you to configure and save multiple configuration files in the same directory. The following example executes the Master TS using the configuration file named MasterConfig.xml  Example: *C:\Program Files\dnp3\Testset.exe –F MasterConfig.xml* |
| ***-S***  This argument specifies that the slave TS will be used. The following example executes the Slave TS using the configuration file named SlaveConfig.xml  Example: *C:\Program Files\dnp3\Testset.exe -S SlaveConfig.xml* |
| ***-G***  This argument generates a new default configuration file (if it doesn’t exist) and then exits. The following examples will create a default configuration files named dnp3\_master\_default.xml and dnp3\_slave\_default.xml respectively.  Example: *C:\Program Files\dnp3\Testset.exe –G*  *C:\Program Files\dnp3\Testset.exe –S –G* |
| ***-E***  This argument is the same as the –G argument except it generates a new default configuration file (if it doesn’t exist) and then executes the TS using that configuration file. |
| ***-L <log level>***  This argument overrides the log level specified by the configuration file. Possible choices are a=ALL, d=DEBUG, i=INFO, c=COMM, p=INTERPRET, w=WARNING, e=ERROR, v=EVENT. The following example executes the Master TS using the configuration file named MasterConfig.xml and sets the log level to INFO.  Example: *C:\Program Files\dnp3\Testset.exe –F MasterConfig.xml –L i* |

# xml Configuration Files

Master and slave configuration files are created when the corresponding master and slave module shortcuts are executed. These configuration files are located in the folder specified during the installation of the TS*.* All DNP parameters required to configure the master and slave modules are located in the xml configuration files. The default master and slave files are named **dnp3\_master\_default.xml** and **dnp3\_slave\_default.xml** respectively.

The TS is fully functional after installation. By default, the configuration files are configured to allow master/slave communication on the local machine via the loopback. The user can become acquainted with issuing commands and monitoring DNP communication using the default configuration files. The list of Test Set commands can be found in the *Test Set Operation* section of this document.

This section details the parameters of each configuration file and their function.

**Master Configuration File**

**Table 1. Master Configuration Parameters**

| Parameter | Description | Value |
| --- | --- | --- |
| PhysicalLayer | Name of the desired physical layer to be used. TCP/IP and Serial parameters are defined in the *TCPClient Name* or *Serial Name* elements. | This name must correspond to the *TCPClient Name* or *Serial Name* attributes specified in the PhysicalLayerList element.  **Default = tcpclient** |
| LogFile | Upon initial startup of the master test set, a log file is created in the test set directory with the name specified in this attribute. | Arbitrary log file name. If “-“ is specified, no log file will be created.  **Default = master\_testset.log** |
| Log Filter | Specifies the desired level of logging upon startup. | Acceptable values are:  **LOG\_EVENT** – Logs only events  **LOG\_ERROR** – Logs errors & events  **LOG\_WARNING** – Logs warning, error & event  **LOG\_INTERPRET** = Logs interpreted hexadecimal DNP line communication, warning, error, & event  **LOG\_COMM** = Logs communication messages, interpret, warning, error, & event  **LOG\_INFO** = Logs information messages, comm, interpret, warning, error, & event  **LOG\_DEBUG** = Logs debugging messages, info, comm, interpret, warning, error, & event  **Default = LOG\_WARNING** |
| LinkLayer.IsMaster | Configures the link layer for master or slave mode, and sets the master/slave bit on all messages accordingly. | true or false  **Default = true** |
| LinkLayer.LocalAddress | DNP address of the local device | **Default = 100** |
| LinkLayer.RemoteAddress | DNP address of the remote device | **Default = 1** |
| LinkLayer.UseConfirmations | If true, the link layer will send data requesting confirmation | **Default = false** |
| LinkLayer.AckTimeoutMS | The response timeout in milliseconds for confirmed requests | **Default = 1000ms** |
| LinkLayer.NumRetries | The number of retry attempts the link will attempt after the initial try. | **Default = 3** |
| AppLayer.Timeout | The application response/confirm timeout in milliseconds. | **Default = 10000ms** |
| AppLayer.MaxFragSize | The maximum allowed size of received application layer fragments. | **Default = 2048** |
| AppLayer.NumRetries | Number of retries performed for applicable frames. | **Default = 3** |
| MasterSettings.AllowTimeSync | When this attribute is set to “true”, the master will perform time syncs when it sees the time IIN bit from the slave | **Default = true** |
| MasterSettings.TaskRetryMS | The time delay in milliseconds between task retries | **Default = 5000ms** |
| MasterSettings.IntegrityPeriodMS | Period in milliseconds for integrity (class 0) scans. | Use -1 to do a single integrity scans on startup  **Default = 60000ms** |
| ExceptionScan ClassX | Determines which data class is returned during an exception scan. Only changed values are returned during an exception scan. | true or false  **Default : true** |
| ExceptionScan PeriodMS | Exception scan rate in milliseconds. | **Default = 2000ms** |
| Unsol.DoTask | If true, the master will enable or disable unsolicited communication based on the **UnsolEnable** bit | **Default = false** |
| Unsol.Enable | Enables or disables unsolicited reporting. The master only uses this attribute if the **UnsolDoTask** bit equals true. | **Default = false** |
| Unsol.ClassX | Determines which data classes are enabled/disabled for unsolicited reporting on master startup | true or false  **Default : true** |
| Physical.TCPClient Name | The name assigned to the TCP client attribute element. Multiple TCPClient elements can be listed in the xml file. The test set uses the communication parameters assigned in the PhysicalLayer attribute. *See Physical Layer* | Arbitrary name assigned to the TCP Client element.  **Default = tcpclient** |
| Physical.TCP OpenRetryMS | Time to wait in milliseconds before trying to reconnect a socket after a connect failure. | **Default = 5000ms** |
| Physical.Address | TCP/IP address of the slave device | **Default = 127.0.0.1** |
| Physical.Port | TCP/IP port of the slave device | **Default = 4999** |
| Physical.Serial Name | The name assigned to the serial client attribute element. Multiple serial elements can be listed in the xml file. The test set uses the communication parameters assigned in the PhysicalLayer attribute. *See Physical Layer* | Arbitrary name assigned to the serial client element.  **Default = serial** |
| Physical.Serial OpenRetryMS | Time to wait in milliseconds before trying to reconnect a serial link after a lost link. | **Default = 5000ms** |
| Physical.Device | Name of the serial port. On Windows these are COMXX, on Linux /dev/ttyXX | Value = ComX | dev/ttyXX  **Default = Com1** |
| Physical.BaudRate | Serial communication baud rate, All rates may not work on all platforms. Check the allowed settings for your platform/port. | Acceptable Values are  1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, & 230400  **Default = 9600** |
| Physical.DBits | Databits | Value = DATABITS\_7  DATABITS\_8  **Default = DATABITS\_8** |
| Physical.StopBits | Stop Bits | Value = STOPBITS\_0  STOPBITS\_1  STOPBITS\_2  **Default = STOPBITS\_1** |
| Physical.Parity | Parity | Value = PARITY\_NONE  PARITY\_ODD  PARITY\_EVEN  **Default = PARITY\_NONE** |
| Physical.FlowControl | Flow Control | Value = FLOW\_NONE  FLOW\_HARDWARE  FLOW\_XONXOFF  **Default = FLOW\_NONE** |

**Slave Configuration File**

The Slave Configuration file is used to configure the slave module to simulate a secondary DNP device or slave, IED, RTU, etc…The following table outlines the configuration of the slave and describes the slave xml file parameters. Refer to Table 1 for a description of the *Link*, *App*, and *Physical* slave parameters not listed Table 2.

**Table 2. Slave Configuration Parameters**

| Parameter | Description | Value |
| --- | --- | --- |
| PhysicalLayer | Name of the desired physical layer to be used. TCP/IP and Serial parameters are defined in the *TCPServer Name* or *Serial Name* elements. | This name must correspond to the *TCPServer Name* or *Serial Name* attributes specified in the PhysicalLayerList element.  **Default = tcpserver** |
| LogFile | Upon initial startup of the slave test set, a log file is created in the test set directory with the name specified in this attribute. | Arbitrary log file name. If “-“ is specified, no log file will be created.  **Default = “-“** |
| Remote | Used for development only | **false** |
| RemotePort | Used for development only | **4998** |
| LinkCommandStatus | Used for development only | **false** |
| StartOnline | User for development only | **false** |
| Log Filter | Same as Master Config File. See Table 1 |  |
| MaxBinaryEvents | The number of binary events the slave will buffer before overflowing | **Default = 100** |
| MaxAnalogEvents | The number of analog events the slave will buffer before overflowing | **Default = 100** |
| MaxCounterEvents | The number of counter events the slave will buffer before overflowing | **Default = 100** |
| TimeIINTask DoTask | If True, the slave will request time synchronization on an interval | true/false  **Default = false** |
| PeriodMS | The period of time synch interval in milliseconds | **Default = 30000ms** |
| UnsolDefaults DoClassX | If true, the corresponding class is enabled for unsolicited messages. | True/false  **Defaults = true** |
| PackDelayMS | The amount of time in milliseconds the slave will wait before sending unsolicited data. | <= 0 == immediate  **Default = 200ms** |
| RetryMS | Time delay in milliseconds that the slave will wait before retrying an unsuccessful unsolicited response | **Default = 2000ms** |
| TCPServer Name | The name assigned to the TCP server attribute element. Multiple TCPServer elements can be listed in the xml file. The test set uses the communication parameters assigned in the PhysicalLayer attribute. *See Physical Layer* | Arbitrary name assigned to the TCP Server element.  **Default = tcpserver** |
| TCP OpenRetryMS | Time to wait in milliseconds before trying to reconnect a socket after a lost link. | **Default = 5000ms** |
| Port | TCP/IP port of the slave device | **Default = 4999** |
| Serial Name | The name assigned to the serial server attribute element. Multiple serial elements can be listed in the xml file. The test set uses the communication parameters assigned in the PhysicalLayer attribute. *See Physical Layer* | Arbitrary name assigned to the serial server element.  **Default = serial** |
| Serial OpenRetryMS | Time to wait in milliseconds before trying to reconnect a serial link after a lost link. | **Default = 5000ms** |

**Slave Configuration File – DNP Data Map Configuration**

The slave configuration file contains all of the information required to simulate a secondary DNP device, including the DNP data map. The TS provides a DNP Level 2 implementation. Refer to the DNP3 specification for items supported within a DNP Level 2 device.

The following table provides a description of the attributes used to define the DNP objects that the slave module will provide to a DNP master as well as the attributes used to define the DNP data map. Refer to the DNP Slave xml file for a DNP data map examples.

**Table 3. DNP Data Map Parameters**

| Parameter | Description | Value |
| --- | --- | --- |
| Static BinaryGroupVar | The default DNP3 group/variation to use for static binary responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 1**  **Var = 2** |
| Static AnalogGroupVar | The default DNP3 group/variation to use for static analog responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 30**  **Var = 1** |
| Static CounterGroupVar | The default DNP3 group/variation to use for static counter responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 20**  **Var = 1** |
| SetpointStatusGroupVar | The default DNP3 group/variation to use for static setpoint status responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 40**  **Var = 1** |
| Event BinaryGroupVar | The default DNP3 group/variation to use for event binary responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 2**  **Var = 1** |
| Event AnalogGroupVar | The default DNP3 group/variation to use for event analog responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 32**  **Var = 1** |
| Event CounterGroupVar | The default DNP3 group/variation to use for event counter responses. | DNP3 Level 2 implementation objects.  **Default: Grp = 22**  **Var = 1** |
| TemplateName | Arbitrary name for the DNP data map. | **Default = default** |
| Index | DNP index # for the associated data type. Multiple indexes can be listed for each data type including Binary, Analog, Counter, Setpoint, Control, Setpoint Status, Control Status |  |
| Name | Arbitrary Name for the associated DNP point. |  |
| ClassGroup | DNP class assigned to the associated DNP point. | **Class, 1, 2, & 3**  **Default = Class 1** |
| DeadBand | Value used for analogs only. Value of analog change for which nothing is reported by the slave. Used to filter small changes in the analog value. | **Default = 0** |
| ControlMode | DNP control type.(Used for set points and controls only) | Value = SBO  DO\_ONLY  SBO\_OR\_DO  **Default = SBO** |
| SelectTimeoutMS | Used for SBO controls and set points. Control action will timeout if the delay between select and control is larger than this value (milliseconds). | **Default = 5000ms** |

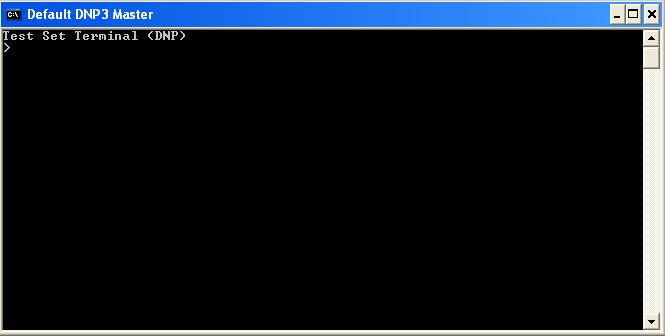
# 

# Executing the TS Master and Slave

The TS installer installs the Test Set application and creates both a master and slave shortcut in the Windows Start Menu. Default xml configuration files are created for the master and slave when each when the master and slave modules are executed from the Windows Start Menu. The default location for the TS Files is *C:\Program Files\dnp3.*  The xml configuration files must be located in the same directory as the testset.exe file.

To execute the Master and Slave, select Start => DNP3 => Default DNP3 Master or Default DNP3 Slave. When executed the TS command line interface appears for both the Master and Slave.

**Fig 1. Master Test Set Interface**



As mentioned previously the TS is fully functional after installation, by default the configuration files are configured to allow master/slave communication on the local machine. The user can become acquainted with issuing commands and monitoring DNP communication using the default configuration files.

# Test Set Operation

The Master and Slave test sets are both command line driven. This section lists each of the commands, subcommands, and their respective function.

**Master Test Set Commands**

* **help** – displays information for commands and subtopics
* ***clear***
* ***screen –*** Clears the test set interface screen
* ***dump***
  + ***start [filename]*** – Starts a log file with the given file name in the test set directory.
  + ***stop –*** Stops the TS from logging to a file.
* **echo**
  + <argument> - echoes the argument to the screen and to the log file. This is helpful for creating annotated log files.
* **print**
  + ***log [device name]*** – Prints the current contents of the log to the specified device.
  + ***loggers*** – Prints out all registered logger names
  + ***vars*** – Prints non-operational information to the console
* **quit** – Closes the DNP Master Application
* **run**
  + ***log*** – Continuously outputs log entries to the screen as they occur. (press *Enter* to stop)
  + ***show*** – Displays data every time it changes???
* **show** – Displays all DNP data on the screen.
* **set**
  + ***level [a:d:i:c:p:w:e:v:n] <logger1> <logger2>***…

A=ALL, d=DEBUG, i=INFO, c=COMM, p=INTERPRET, w=WARNING, e=ERROR, v=EVENT

Used to set or clear log level filtering. If no loggers are specified, all loggers are affected.

* + ***logcol [t:f:d:l:m]***

T=Time, f=Filter, d=Device, l=Location, m=Message

Used to set the order/presence of columns for displaying log entries.

* + ***Show <all:bi:ai:c:cs:ss> <index start> <index stop>***

Bi=Binary Input, ai=Analog Input, c=Control, cs=Control Status, ss=Setpoint Status

Used to display a range when the show command is executed.

* **issue**
  + ***bo <index> <code> <on time> <off time> <count>***

**code**= pon (pulse on), lon, (latch on), loff(latch off), pclose(relay output close), ptrip, (relay output trip)

**on time** = pulse on time (not required for lon & loff. Default =100ms)

**off time** = pulse off time (not required for lon & loff. Default =100ms)

**count** = number of control operations to perform (Default = 1)

Used to issue a binary output

* + ***st <index> <value>***

Used to issue a setpoint output

**Slave Test Set Commands**

* **help** – displays information for commands and subtopics
* ***clear***
* ***screen –*** Clears the test set interface screen
* ***dump***
  + ***start [filename]*** – Starts a log file with the given file name in the test set directory.
  + ***stop –*** Stops the TS from logging to a file.
* **echo**
  + <argument> - echoes the argument to the screen and to the log file. This is helpful for creating annotated log files.
* **print**
  + ***log [device name]*** – Prints the current contents of the log to the specified device.
  + ***loggers*** – Prints out all registered logger names
  + ***vars*** – Prints non-operational information to the console
* **quit** – Closes the DNP Master Application
* **run**
  + ***log*** – Continuously outputs log entries to the screen as they occur. (press *Enter* to stop)
* **set**
  + ***level [a:d:i:c:p:w:e:v:n] <logger1> <logger2>***…

A=ALL, d=DEBUG, i=INFO, c=COMM, p=INTERPRET, w=WARNING, e=ERROR, v=EVENT

Used to set or clear log level filtering. If no loggers are specified, all loggers are affected.

* + ***logcol [t:f:d:l:m]***

T=Time, f=Filter, d=Device, l=Location, m=Message

Used to set the order of columns for displaying log entries.

* + ***response [all:bo:st] [all:index#] [code]***

Code=SUCCESS, TIMEOUT, NO\_SELECT, FORMAT\_ERROR, NOT\_SUPPORTED, ALREADY\_ACTIVE, HARDWARE\_ERROR, LOCAL, TOO\_MANY\_OPS, NOT\_AUTHORIZED

Used to set the response code that the slave returns for incoming binary output (bo) and setpoint (st) commands.

* **queue**
  + ***ai <index> <value> <quality\*>***

Used to queue an analog input value into the transaction buffer.

* + ***bi <index> <value> <quality>***

Used to queue a binary input value (0 or 1) into the transaction buffer.

* + ***c <index> <value> <quality>***

Used to queue a counter value into the transaction buffer.

*\*Quality codes can be generated using the flag octet information for each DNP3 object. Refer to the DNP3 Specification for more information.*

* **flush**– Flushes the data in the transaction buffer to the slave stack. (Used with the queue command)