

The NetDAQ Developer's Toolbox

Technical Data

The NetDAQ Developer's Toolbox consists of two Windows® Dynamic Link Libraries (DLLs) which allow users to create custom Windows application programs to communicate with one or more NetDAQ instruments. The Developer's Toolbox enables users to automate or customize their use of NetDAQ, yet take advantage of functionality already provided by NetDAQ Logger for Windows.

- Create custom user interfaces for NetDAQ applications.
- Access real-time NetDAQ data and store it in any desired format, such as a custom database.
- Load different NetDAQ setup files.
- Change Mx+B values for each channel in a NetDAQ instrument.
- Start and stop one or multiple NetDAQ instruments.
- Control NetDAQ's digital I/O channels.
- Access and control NetDAQ's serial port.

The Developer's Toolbox runs on the Microsoft® Windows, Win95® and WinNT® operating systems and allows for development of custom NetDAQ applications with either Microsoft's Visual BASIC, or the C and C++ programming languages.

Functional overview

The Developer's Toolbox is a set of routines which manipulates NetDAQ measurement hardware through NetDAQ Logger for Windows application software. Most Developer's Toolbox functions are accomplished by posting

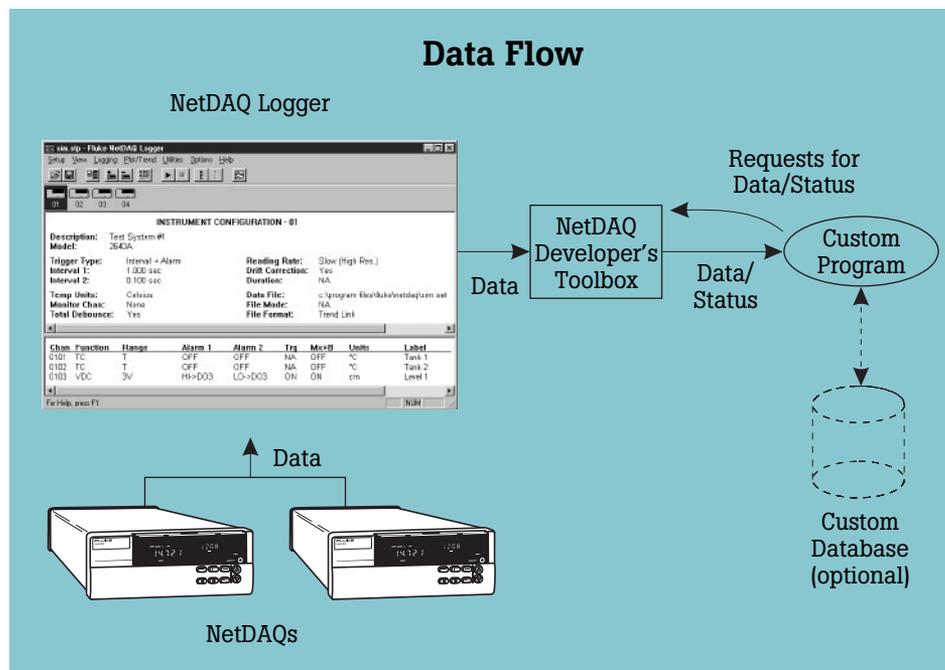
messages to NetDAQ Logger, causing it to perform tasks that would otherwise be initiated by interactive mouse or keyboard input. NetDAQ Logger must be running while the Developer's Toolbox is in use, but it can be minimized (reduced to an icon) so as not to interfere with the user interface of the custom application.

The functionality provided by the Developer's Toolbox can be categorized as follows:

Scan retrieval: For each active NetDAQ instrument, the Developer's Toolbox can receive and queue scan data from NetDAQ Logger. The application program

can retrieve one or all of the queued scans, or query for the most recent scan. Each scan contains a time stamp, alarm state, DIO state, totalizer value, and readings for each configured channel of the instrument. These functions can also provide status about the queue of scans for each instrument, or clear the queue.

NetDAQ Logger control: The Developer's Toolbox provides control of essential NetDAQ Logger functionality, such as starting and stopping scanning, and loading a NetDAQ Logger setup file or Mx+B scaling information.



With the Developer's Toolbox, data flows from the NetDAQ instruments, through NetDAQ Logger for Windows, through the Developer's Toolbox, to the custom application program which then stores the data in a custom format or database.

Status and information: The Developer's Toolbox provides information about instruments' scanning status, the status of their scan queues, alarms status, NetDAQ Logger communications errors, etc. It also provides a list of instruments currently configured in NetDAQ Logger, and channel configuration information for producing data reports.

Serial Pass-Thru: The Developer's Toolbox provides the ability to send and receive ASCII data through a NetDAQ instrument's RS-232 serial port. Functions in the Developer's Toolbox can instruct a NetDAQ instrument to send ASCII data out its RS-232 port, and can retrieve ASCII data which the instrument has received and buffered. This can be used to monitor data and/or send commands to serial devices attached to the instrument's RS-232 port, such as analog output modules or ASCII terminals.

Miscellaneous: The Developer's Toolbox provides the ability to set one or more DIO lines on a NetDAQ instrument and to clear the instrument's totalizer.

Applications

Remote response: The Developer's Toolbox allows a customer's application to generate automated, non-time-critical

responses to scan data coming from NetDAQ instruments. The application can monitor the scan data for a particular condition and then activate a digital alarm or analog output module either by setting NetDAQ's DIO lines or by sending characters through its serial port.

Alternative user interface: In some situations, there may be a need for a more customized or familiar user interface than that provided by NetDAQ Logger; or it may be important to restrict an operator's interactions with NetDAQ Logger to prevent unexpected alterations to instrument configurations or the scan retrieval process. With the Developer's Toolbox, custom applications can be developed with any desired user interface. The user interface could be as simple as one with just a start and stop button, while still performing all the necessary operations.

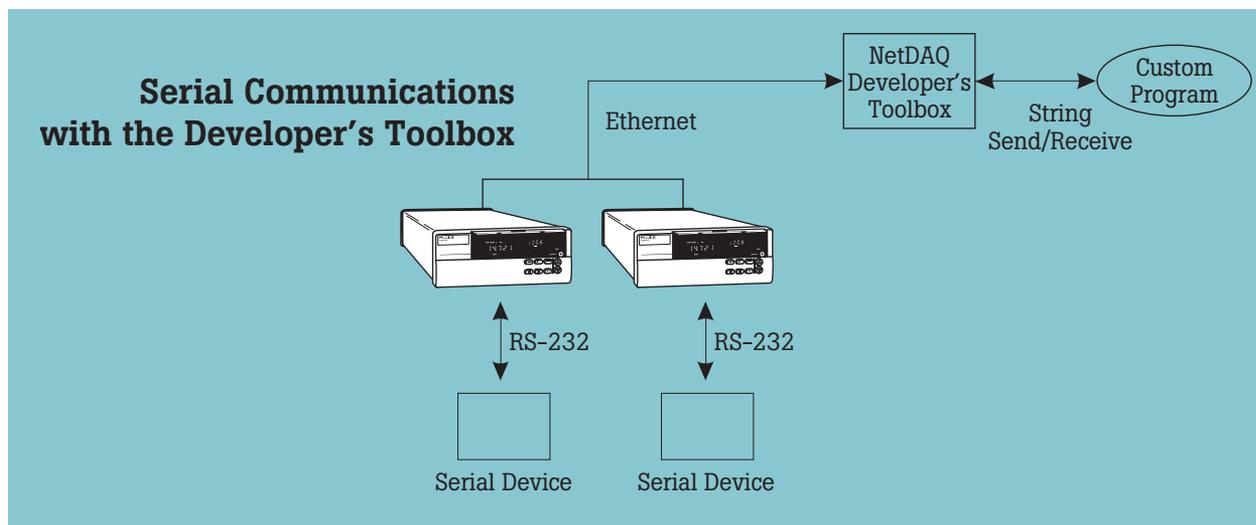
Alternative data storage formats: The Developer's Toolbox allows for real time NetDAQ data to be stored in customized formats other than the ones provided by NetDAQ Logger (CSV, Trend Link, or NetDAQ's binary format). This is done by extracting scans from NetDAQ instruments and storing the data in custom formats or databases.

Serial Pass-Thru

The Developer's Toolbox provides the ability to send and receive ASCII data through a NetDAQ instrument's RS-232 serial port. The Serial Pass-Thru mode is general purpose, allowing for communication with any RS-232 device, such as:

- Analog output modules
- Digital I/O modules
- ASCII terminals
- Lights or horns
- Sensors which output ASCII data, for example: humidity sensors, optical pickups, etc.

When using the Serial Pass-Thru mode, a NetDAQ instrument is equivalent to an ethernet-to-serial converter. Functions in the Developer's Toolbox can instruct a NetDAQ instrument to send ASCII data out its RS-232 port, and can retrieve ASCII data (up to 2000 characters at a time) which the instrument has received and buffered. However, this mode does not allow for the control of a NetDAQ instrument via the serial port. All control of NetDAQ instruments must be done through the Ethernet interface via NetDAQ Logger or the Developer's Toolbox.



The Serial Pass-Thru mode turns a NetDAQ instrument into an Ethernet-to-Serial converter.

Function listings

The Developer's Toolbox functions can be divided into five areas: scan retrieval, NetDAQ Logger control, instrument status and information, Serial Pass-Thru, and miscellaneous functions. The following sections list and describe each of the different functions.

Scan retrieval functions

These functions are used for extracting scan data stored in the Developer's Toolbox by NetDAQ Logger. The Developer's Toolbox maintains a scan queue for each active instrument. Each scan contains a time stamp, alarm state, DIO state, totalizer value, and readings for each configured channel of the instrument. These functions also clear and provide information about the scan queues.

DbxScansGet

Retrieves the oldest scan from the Developer's Toolbox scan queue and deletes it from the queue.

DbxScansGetLast

Retrieves the most recent scan from the Developer's Toolbox scan queue; does not delete scan from queue. Useful for polling for latest scan data.

DbxScansGetLastChan

Same as DbxScansGetLast(), but returns a reading for a channel, instead of the entire scan.

DbxScansGetLastAlarm

Same as DbxScansGetLast(), but returns the state of instrument's alarms, instead of the entire scan.

DbxScansGetLastDIO

Same as DbxScansGetLast(), but returns the state of instrument's Digital I/O lines, instead of the entire scan.

DbxScansGetLastTotalizer

Same as DbxScansGetLast(), but returns the latest value of the instrument, instead of an entire scan.

DbxScansGetC

Retrieves the oldest scans from the Developer's Toolbox scan

queue and deletes them from the queue. Cannot be called from Visual Basic; used for extracting data from the Developer's Toolbox with highest throughput. This is designed to be used with custom applications written in C and C++, not in Visual Basic.

DbxScansInQueue

Returns the number of scans currently in the Developer's Toolbox scan queue.

DbxScansInInstrument

Returns the number of scans currently in the instrument's scan queue.

DbxScansClear

Clears all scans from the Developer's Toolbox scan queue.

DbxScansBufferSize

Size of the Developer's Toolbox scan queue in scans and bytes; used in conjunction with DbxScansGetC().

DbxMakeLoggerSleepOnNewScans

Causes NetDAQ Logger to sleep when it attempts to record scans into the scan queue of any instrument. This provides a means of calling several Developer's Toolbox functions without NetDAQ Logger adding new scans between calls, (e.g. DbxScansGetLastChan and DbxScansGetLastAlarms).

DbxEndSleepOnNewScans

Allows NetDAQ Logger to wake up and store scans into instrument scan queues.

NetDAQ Logger control functions

These functions allow the custom application to initiate commands on NetDAQ Logger, such as starting and stopping logging, and loading a setup file or a file containing Mx+B values.

DbxCmdLoggingStart

Sends a message to NetDAQ Logger to start logging on an instrument.

DbxCmdLoggingStop

Sends a message to NetDAQ Logger to stop logging on an instrument.

DbxCmdLoggingStartAll

Sends a message to NetDAQ Logger to start logging on all instruments.

DbxCmdLoggingStopAll

Sends a message to NetDAQ Logger to stop logging on all instruments.

DbxCmdLoadSetup

Sends a message to NetDAQ Logger to load a NetDAQ Logger setup file. (All instruments must have stopped logging before this command can be sent.)

DbxCmdLoadMxb

Sends a message to NetDAQ Logger to load a file of Mx+B values; this will change the Mx+B values for an instrument.

DbxCmdLoggerSilent

Sets NetDAQ Logger to work in "silent" mode (no dialog boxes brought up for communications interruptions, etc.).

DbxCmdCompleted

Used to determine when a previous NetDAQ Logger Command has completed.

Instrument status and information functions

These functions provide information about instrument's scanning status, the status of their scan queues, alarms status, NetDAQ Logger communications errors, etc. They also provide a list of currently configured instruments in NetDAQ Logger, and channel configuration information for producing data reports.

DbxInstStatus

Returns a bitmap of status information about instrument and instrument's scan queue.

DbxInstAlarmStatus

Returns a bitmap indicating cumulative alarms triggered since the last call to this function; used in conjunction with DbxInstStatus.

DbxInstPostStatus

Sets the Developer's Toolbox to post messages to the custom application when new scans arrive, communications errors occur, NetDAQ Logger commands complete, etc. This is designed

to be used with custom applications written in C and C++, not in Visual Basic.

DbxInstErrorLogging

If communications errors occur between NetDAQ Logger and the instrument, an error bitmap is returned by this function indicating the nature of the problem.

DbxInstConfigList

Returns a list of the instruments currently configured in NetDAQ Logger.

DbxInstChannelList

Lists the currently configured channels for an instrument.

DbxInstChannelConfig

Returns several strings with channel configuration (reading type, range, units and the channel label from NetDAQ Logger) for an instrument's channel. Used for labeling stored data or creating reports.

DbxInstSetupFile

Returns the name of the setup file (if any) that NetDAQ Logger has loaded.

Serial Pass-Thru functions

The Developer's Toolbox provides the ability to send and receive ASCII data through a NetDAQ instrument's RS-232 serial port. Functions in the Developer's Toolbox can instruct a NetDAQ instrument to send ASCII data out its RS-232 port, and can retrieve ASCII data which the instrument has received and buffered. This can be used to monitor data and/or send commands to serial devices attached to the instrument's RS-232 port.

DbxSerialEnable

Enables instrument's RS-232 port for Serial Pass-Thru.

DbxSerialDisable

Returns instrument's RS-232 port back to normal command.

DbxSerialSend

Sends ASCII data via the instrument's RS-232 serial port.

DbxSerialRecv

Receives ASCII data from the serial data buffer in the instrument.

DbxSerialStatus

Polls status of instrument's serial data buffer.

Miscellaneous Developer's Toolbox Functions

DbxSetDIO

Sets the digital output lines on the instrument.

DbxSetDIOLine

Sets a single digital output line on the instrument.

DbxClearTotalizer

Clears the totalizer counter on the instrument.

Requirements

- NetDAQ Logger software version 2.0 or greater (Model 2640A-911 or 2640A-912).
- Operating system: Microsoft Windows 3.1, Windows for Workgroups 3.1, Windows95 or WindowsNT.
- Floppy disk drive: 1.44 MB (3.5") floppy disk drive.
- Programming language: Microsoft Visual BASIC for Windows 4.0, or any version of C/C++ targeted for Windows 3.1, Windows for Workgroups 3.1, Windows95 or WindowsNT.

Ordering information

Application software

264XA-903	NetDAQ Developer's Toolbox
2640A-911	NetDAQ Logger
2640A-912	NetDAQ Logger with Trend Link

Fluke. *Keeping your world up and running.*®

Fluke Corporation

PO Box 9090, Everett, WA 98206 U.S.A.

Fluke Europe B.V.

PO Box 1186, 5602 BD Eindhoven, The Netherlands

For more information call:

In the U.S.A. (800) 443-5853 or Fax (425) 446-5116
 In Europe/M-East/Africa +31 (0) 40 2675 200 or Fax +31 (0) 40 2675 222
 In Canada (800)-36-FLUKE or Fax (905) 890-6866
 From other countries +1 (425) 446-5500 or Fax +1 (425) 446-5116
 Web access: <http://www.fluke.com>

©1997-2010 Fluke Corporation. Specifications subject to change without notice. Microsoft Windows, Windows95 and WindowsNT are registered trademarks of Microsoft Corporation. Printed in U.S.A. 2/2010 1260272C D-EN-N

Modification of this document is not permitted without written permission from Fluke Corporation.