CDOAN-Script

DNP3 Protocol Tester – Script Editor
Release 1.1
June 17, 2024

www.cdoancal.com

T P I L B

Contents

Introduction

This document describes the process for creating a script file with a series of commands to be executed by CDOAN-DNP3 operating in master simulation mode. The set of script commands, as described in the following pages, are saved to a text file. The commands are run by specifying the file name in a “script” entry in the Master Station Configuration View.

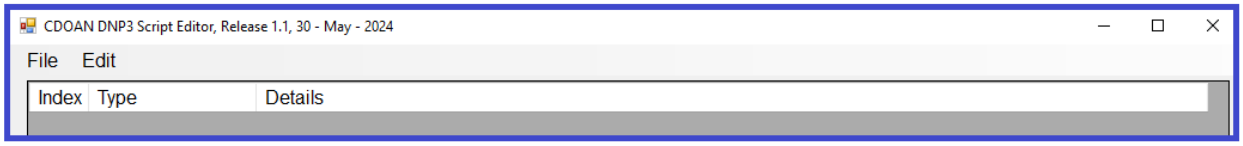
For example, the following entry in CDOAN-DNP3, Master view:

causes instructions in the script file “C:\temp\test.txt” to be processed, with a message transmitted once every 2 seconds to destination outstation index 1.

Main View

The script editor is installed as program CDOAN-Script and can be found under the CDOAN group in the start menu.

The main view of the script editor shows the list of commands created. An empty script file is shown as:

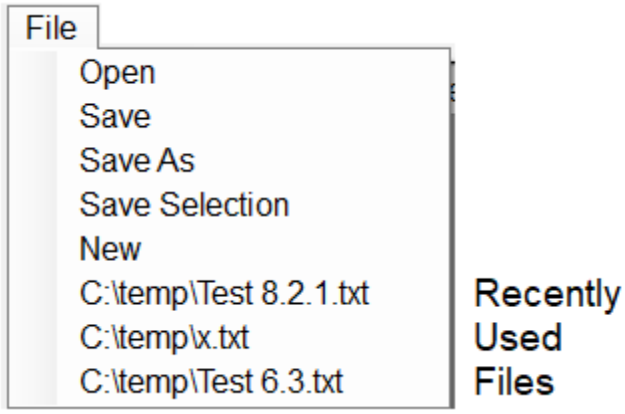


This is the starting point. From here, the user creates a sequence of commands that are eventually saved to a script file (.txt), to be eventually read and processed by CDOAN-DNP3. While the script file is in .txt format, it cannot be edited by any program other than the CDOAN Script editor. Or, to be more precise, it can be edited but CDOAN-DNP3 will not process a file edited by any other program. This protection is placed to guarantee the integrity of the script file in that consistency checks, performed by the editor during the save process, remain valid.

The CDOAN-Script program has two pull-down lists, ***File*** and ***Edit***.

- The ***File*** pull-down contains standard operations to open and save files.
- The ***Edit*** pull-down provides the ability to create, modify, and delete script command entries

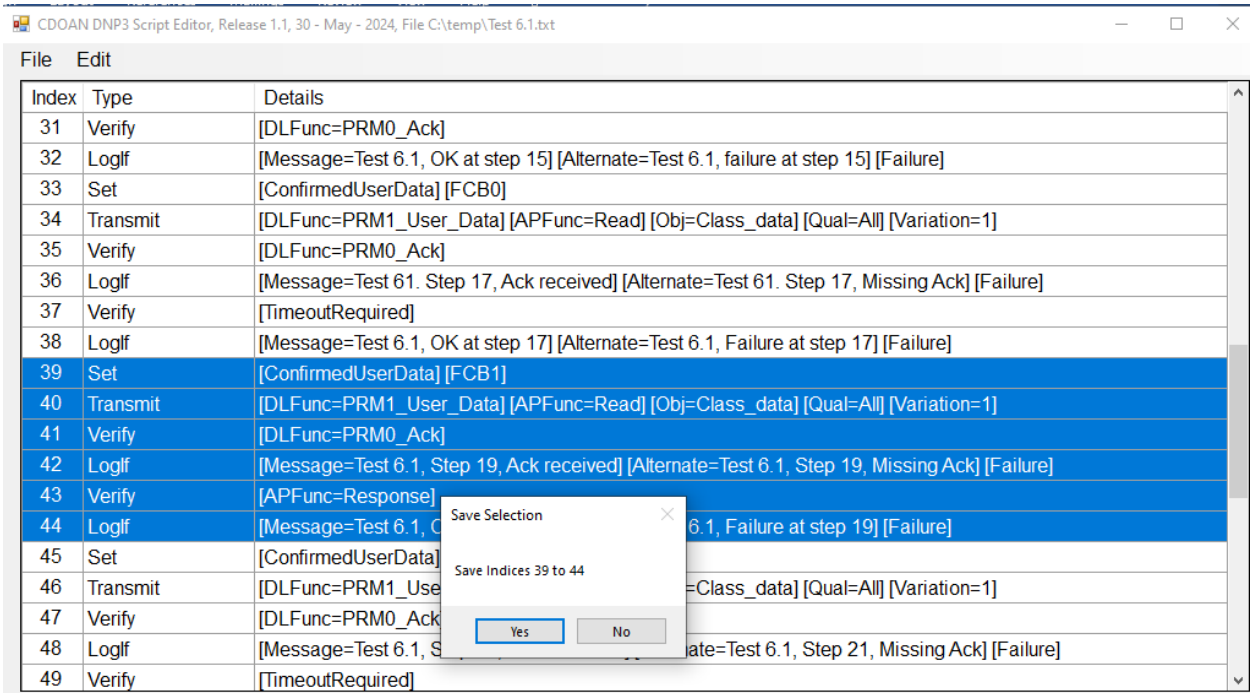
File Pull-Down List



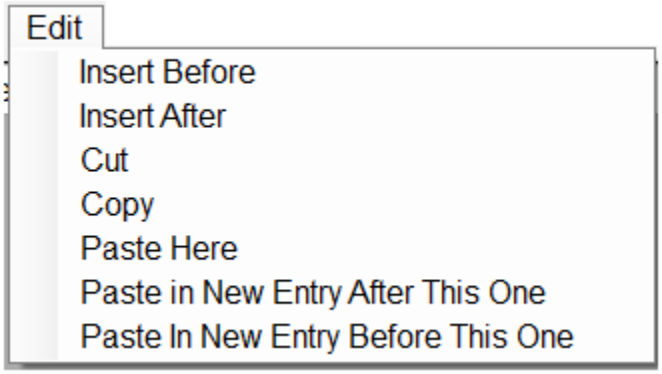
OPEN	Opens an existing script file
SAVE	Saves the current contents to the most recently opened or saved file
SAVE AS	Saves the current contents to a new file
SAVE SELECTION	Saves a selected range of script entries to a new file
NEW	Erases all current script entries

The three most recently used files are shown in the bottom of the pull-down menu. Clicking on any of these entries opens the corresponding file.

Sample screen shot for SAVE SELECTION.



Edit Pull-Down List



INSERT BEFORE	Inserts a new script entry before the active entry
INSERT AFTER	Inserts a new script entry after the active entry
	When there are no entries (as for a new file) INSERT BEFORE and INSERT AFTER are identical
CUT	Removes the current entry
COPY	Makes a copy of the current entry
PASTE HERE	Replaces the current entry with the most recently CUT or COPY entry
PASTE AFTER	Inserts the most recently CUT or COPY entry as a new entry after the current entry
PASTE BEFORE	Inserts the most recently CUT or COPY entry as a new entry before the current entry

Existing entries can be edited by double-clicking on the field to edit (or right-click).

Script Commands

Selecting *Edit/Insert Before* or *Edit/Insert After* presents a list of all possible script commands for selection. Commands are of six types:

Type	*Commands*			
Message Transmission	Transmit	AddObject		
Evaluate the success or failure of an outstation response	Verify	And	Or	
Log Information	Log	LogIf	Prompt	
Flow Control	Restart	Stop	Repeat	Until
	If	Else	Endif	Wait
Parameters and Options	Set	Variable	ForceError	
Debugging. Use only when instructed to do so	Trace			

Philosophy

Before creating commands, the user should know the philosophy of script execution.

- A script file is a series of commands
- Commands in the script are generally executed sequentially, but several commands exist to alter this flow
- The main purpose of a script is to send messages to an outstation and analyze the responses
- At any given time, the script exists in a success or failure state. The state is "success" at startup. Once a state is assigned, it remains unchanged until a new state is assigned. That is, nothing implicitly changes the state
- ***Verify***, ***Or***, ***And***, and ***Prompt*** commands are the only ones that may change the script state. ***Verify***, ***Or***, ***And*** are all defined identically, and include choices to select different parts of the outstation response message to evaluate. Any subset of these choices can be selected. The result of the evaluation is "success" if tests for all selected choices pass, or "failure" if at least one test fails.
    - ***Verify*** sets the script state to the result of the evaluation
    - ***Or*** sets the script state to success if the result of the evaluation is success. The script state is unchanged if the result of the evaluation is failure
    - ***And*** sets the script state to failure if the result of the evaluation is failure. The script state is unchanged if the result of the evaluation is success

  Prompt asks the user a question. Answering ***yes*** sets the state to success. Answering ***no*** sets the state to failure.
- Messages can be written to the communication view with the ***Log*** or ***LogIf*** commands. All messages written to the communication view are also written to the ChangeEvents view
    - ***Log*** write a message unconditionally
    - ***LogIf*** writes one message if the current script state is success, or a separate message if the state is failure. The command can also be used to stop the script after logging either the success or failure message
- Script flow commands include:
    - ***Restart*** to restart the script from the first entry
    - ***Stop*** to stop
    - ***Repeat/Until*** to repeat a set of commands until a condition is met
    - ***If/Else/Endif*** to conditionally execute commands
    - ***Wait*** to pause for a specified time
- Parameters and Options
    - ***Set*** is used to set selected DNP3 protocol options related to confirmed user data
    - ***ForceError*** instructs the script to enable or disable certain DNP3 protocol error conditions
    - ***Variable*** is used to assign, increment, or decrement values associated with any of 26 variables

Messages Transmission Commands

Transmit

Used to configure a message for transmission.

Messages can be defined to transmit data link functions:

- User Data, both unconfirmed and confirmed services
- Request Link Status
- Reset Link States
- NACK
- ACK messages are not configured, but are automatically sent when appropriate (unless explicitly disabled).

Application layer functions include:

- Read
  - Device attributes
  - Binary and double binary static and event
  - Analog static and event
  - Analog deadband
  - Counter and frozen counter static and event
  - Binary and analog output status static
  - Class data
- Write
  - Analog Deadbands
  - Internal Indications
- Commands
  - SBO
  - Direct Operate
  - Direct Operate-No Acknowledge
- Freeze and Freeze-and-Clear
- Cold Restart
- Enable/Disable Unsolicited

Object group, variation, qualifier, range, point index, output point value, and CROB information are solicited as appropriate, based on the object selected

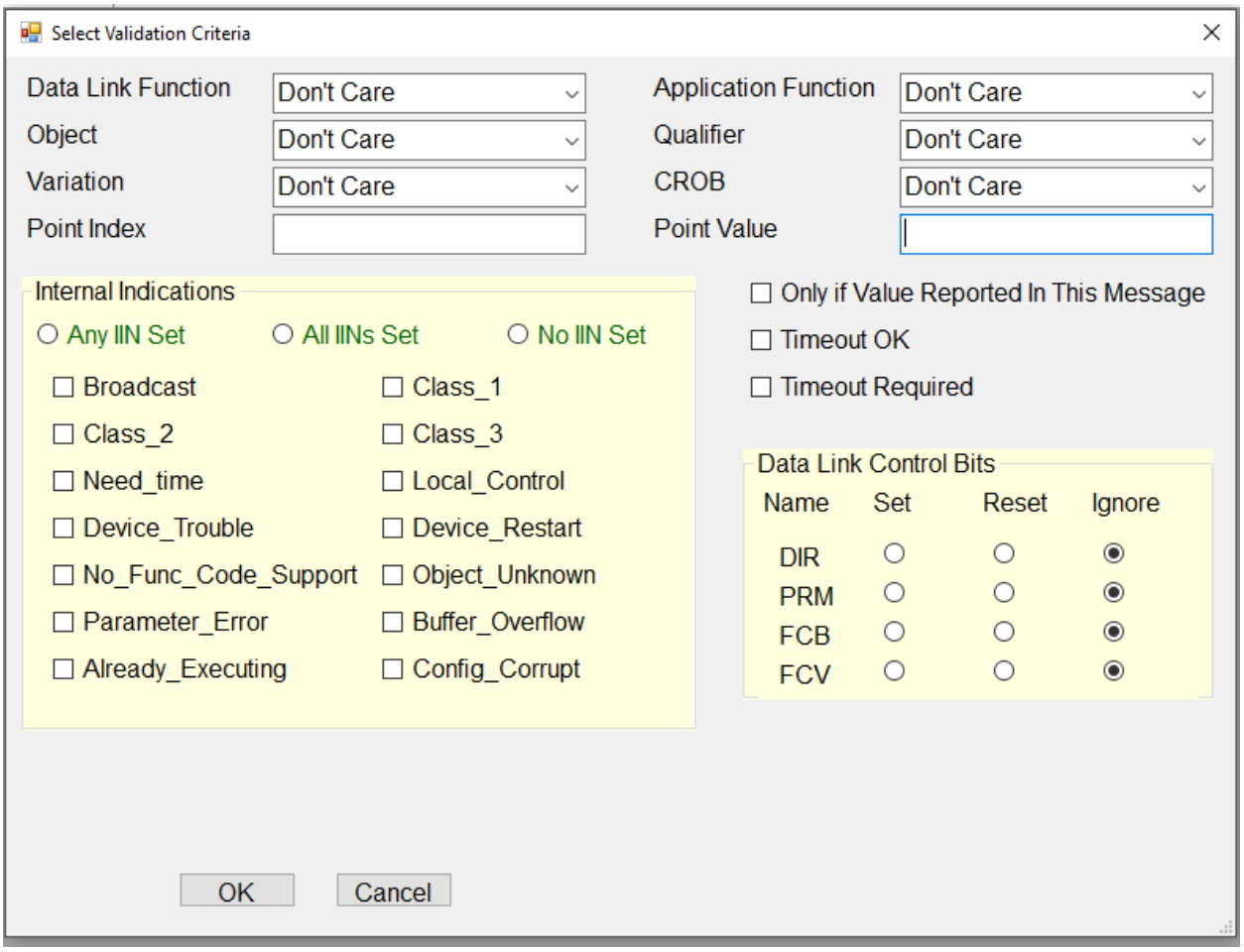
AddReadObject

An ***AddReadObject*** command can be coded after a ***Transmit*** command, only for ***Transmit*** commands with a ***read*** application layer function. Each ***Transmit/read*** entry can only be coded to read a single object. To read more than one object in a single DNP3 message, the second object is coded in an ***AddReadObject*** command, one new object per ***AddReadObject***. The message transmitted includes a read for the object coded in the original ***transmit*** command plus objects in all the following ***AddReadObject*** commands.

Script Evaluation Commands

Verify, And, and ***Or*** commands contain a set of conditions used to evaluate the prior outstation response message. Only those conditions to be evaluated need be selected. The result of the evaluation is success if all selected conditions are true, or failure otherwise.

- For ***Verify***, the script state is set to the result of the evaluation
- For ***Or***, the script state is set to success if the evaluation is successful, but unchanged otherwise
- For ***And***, the script state is set to failure if the evaluation is failure, but unchanged otherwise



- ***Verify***, ***Or***, and ***And*** work on complete application fragments, not individual data link frames.
- When a request is issued that expects two response messages, such as a confirmed user data request with data link acknowledgement and the requested data responses, two ***verify*** commands should be coded, one for the ACK and a second for the data. Either can be followed by Or, And, or LogIf commands, but no others. After any other command is processed, the context of the outstation response message is lost.
- When the ***Object***, ***Point Index***, and ***Point Value*** fields are all coded, the script checks that the specified input point has been reported with the specified value. It is possible that the point was reported in the most recent message received or not. If not, then the value used for verification is the value of the most recent report for the point, possibly several messages ago. When the ***Only if Value Reported In This Message*** box is checked, the comparison is successful only when the point was reported in the current message, and with the correct value. If not checked, then the most recently reported value is used.
- There are two ***Timeout*** options. One when a timeout is OK, the other when it is required

Log Information

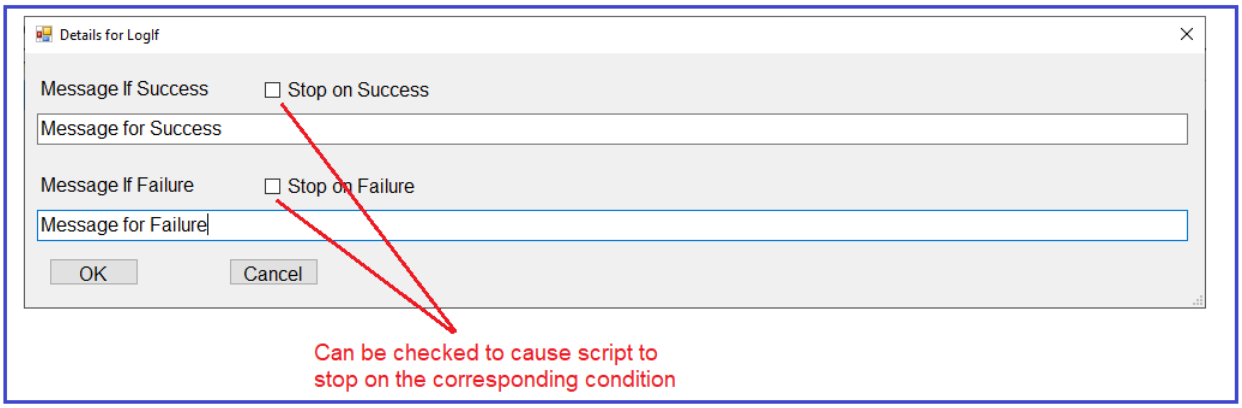
Log

Defines a message to place in the communication view.

The message can be formatted to include current values of script variables. As explained in a following section, the script processor maintains 26 variables named A to Z. Strings $A through $Z, contained in the log message, will be replaced with values of the corresponding variables at log time.

LogIf

Inserts a message in the communication view based on the current script state



Either message can be formatted to include the current values of script variables. As explained in a following section, the script processor maintains 26 variables named A to Z. Strings $A through $Z, contained in the log message, will be replaced with values of the corresponding variables at log time.

Prompt

Asks the user a question, the answer to which instructs the script to continue or not. While the question is posed:

- Communication stops
- If a network-based outstation disconnects and restarts before the prompt is answered, the program reconnects

This command is useful if the outstation is to be restarted and the program should wait until the restart completes.

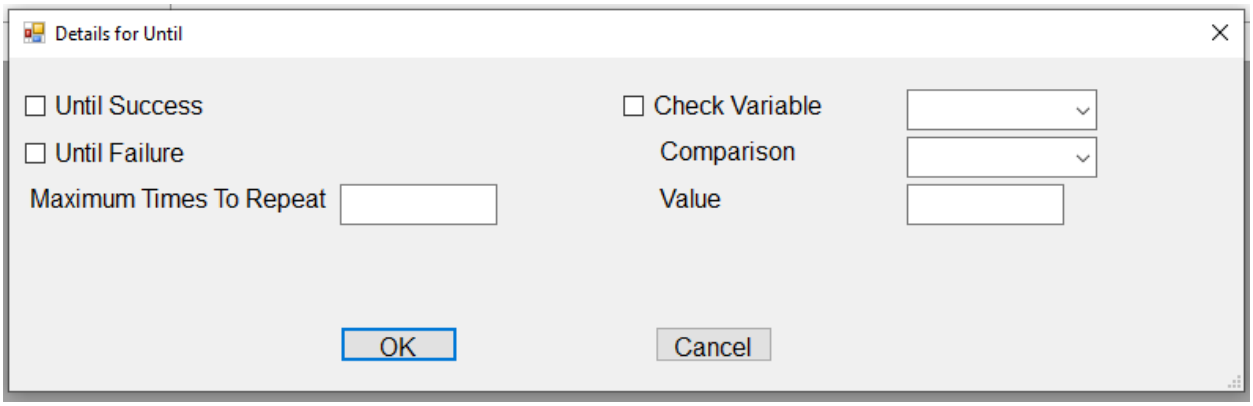
Flow Control

If, Else, EndIf

Alters script flow control based on the current success or failure state

Repeat/Until

Repeats a set of script entries until a condition is met.



The block of commands between the ***Repeat*** and ***Until*** commands are repeated until any of the conditions coded in the ***Until*** command are met. Possibilities are:

- Repeat for a given number of times
- Repeat until script success
- Repeat until script failure
- Repeat until a variable meets or exceeds a given value

Other Flow Control

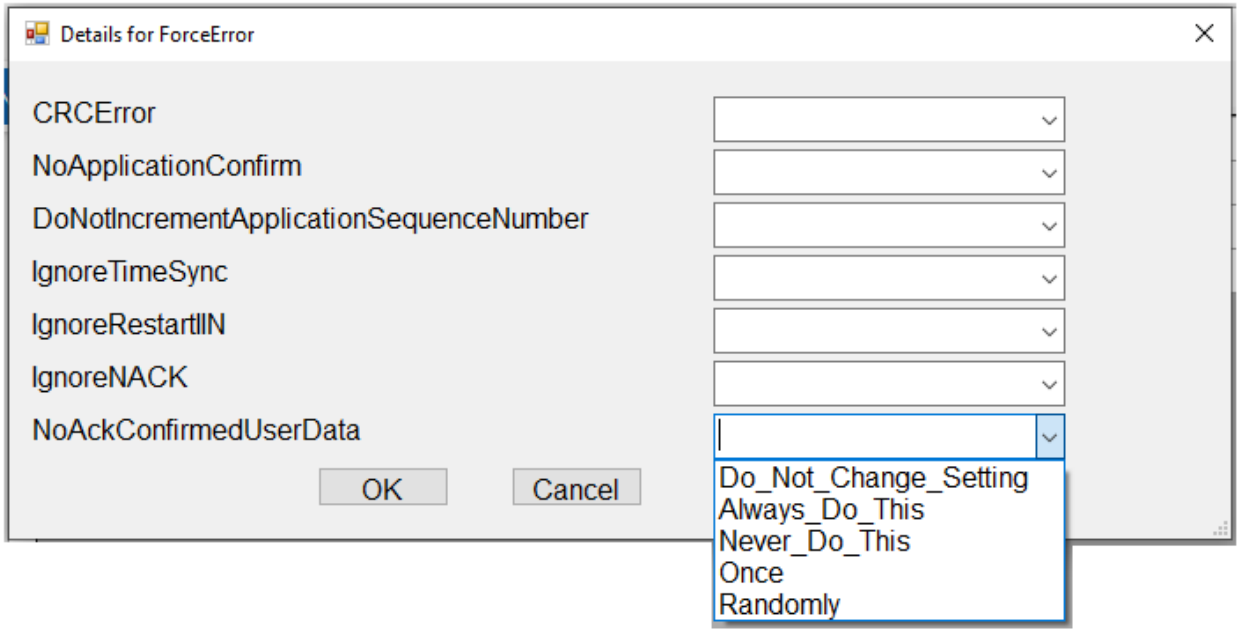
STOP	Stops execution
RESTART	Restarts execution at the first entry
WAIT	Delays execution for a specified number of seconds

Parameters and Options

ForceError

This command supports enabling or disabling a set of DNP3 error conditions. They are:

- Send messages with a CRC error
- Do not send an application confirm when requested
- Do not increment the application sequence number on successive messages
- Ignore outstation requests for time synchronization
- Ignore outstation notifications that it has restarted
- Ignore NACK responses from outstation by not sending subsequent Link Reset messagesHow ablout



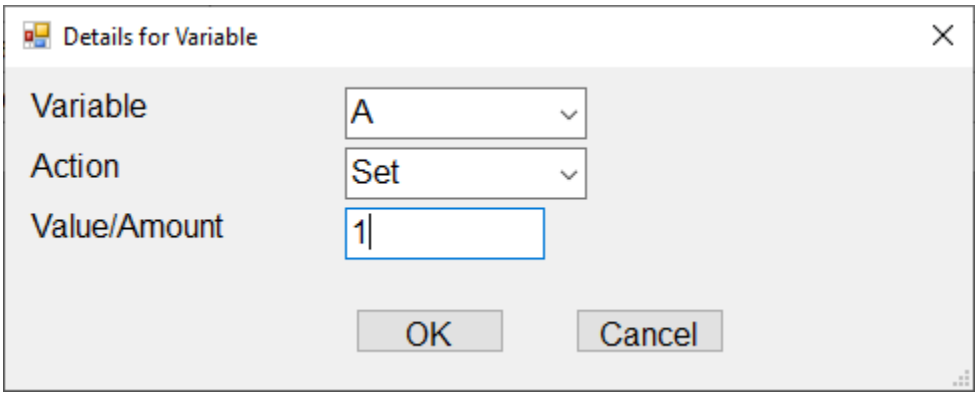
Each condition can be set to occur, either

- All the time
- Never (used to disable an error condition that had been enabled in a previous command)
- Cause the error to occur only one time
- Cause the error conditions to occur randomly

Variable

The script editor supports 26 variables, named A through Z. This script command allows the value of any variable to be set, incremented, or decremented. A variable name can be used in other commands as a:

- point index
- analog output value
- binary output pulse duration
- condition to stop or continue a repeat/until loop



Set

Used to set a variety of DNP3 protocol options. Currently, all options deal with confirmed user data.