

CDOAN-Script

DNP3 Protocol Tester – Script Editor
Release 1.0.0
May 30, 2024

www.cdoanca.com

TPILB

Contents

- Introduction1
- Main View2
- Script Commands3
 - Philosophy3
 - Commands4
 - Communication Script Type.....4
 - Verify, OR, And Command Types.....5
 - Enable/Disable5
 - Log, Logf, and Prompt.....6
 - If, Else, EndIf6
 - Repeat/Until6
 - ForceError6
 - Variable.....6
 - Wait/Stop/Restart6

Introduction

This document describes the process for creating an external script file to be run by CDOAN-DNP3 operating in master simulation mode. A script is a series of commands as described in the following pages. Script commands are saved in a text file, which can then be specified as an entry in the Master Station Configuration View of the CDOAN-DNP3 protocol exerciser.

For example, the following entry in CDOAN-DNP3

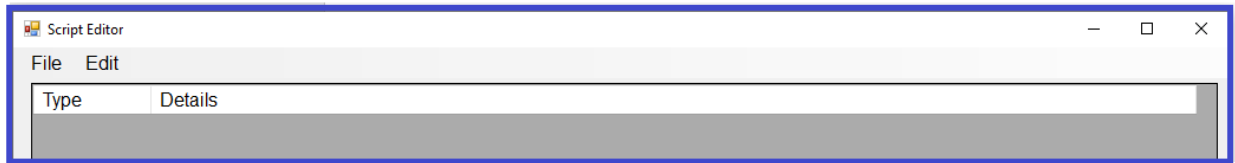
Type	Request	Var	Dest	Per	Frequency
SCRIPT	External	C:\temp\x.txt	1	2	Periodic

Causes messages in the script file "C:\temp\x.txt" to be transmitted once every 2 seconds to outstation index 1.

Script entries are defined in CDAON-Script. The rest of document refers to operation of that program.

Main View

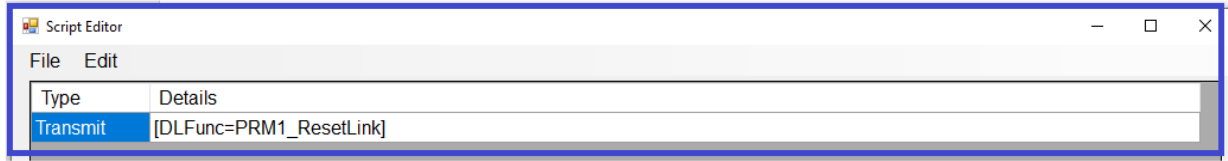
The main view of the script editor is:



This view allows the user to create a script file with a series of commands and to save those commands to an external script file (.txt). This file can be read and processed by CDOAN-DNP3. While the script file is in .txt format, it cannot be edited once created. Or, to be more precise, it can be edited but CDOAN-DNP3 will not process a file edited after being saved by the script editor. During the save process, the editor performs consistency checks that may cause some fields to be changed.

Within CDOAN-Script, there are two pull-down lists. The file pull-down contains standard operations to open and save files. Script entries are created and edited from the edit pull-down list.

Script Commands



The edit pull-down list allows the user to enter a series of script commands. Normal cut, copy, and paste commands exist as in most other editing products. Commands are created by two edit options:

- **Insert Before** inserts a new entry immediately before (above) the currently highlighted entry, and
- **Insert After** inserts a new entry immediately after (below) the currently highlighted entry

Philosophy

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

- The main purpose of a script is to send a sequence of messages to an outstation.
- A script file includes a series of commands, some of which define messages to be sent.
- Commands in the script are generally executed sequentially, but the flow can be changed based on the success or failure of prior commands.
- At any given time, the script has a success or failure status. The status is “success” at startup.
- The status can be changed by a “Verify” command, which tests conditions within an outstation response message. The status is set to “success” if all tests pass. The status is set to “failure” if one or more test fails.
- There is also an “Or” command, which is similar to the “Verify” statement. The status is set to “success” if all tests pass. The status is unchanged if one or more tests fail.
- Similarly, there is an “And” command. The status is unchanged if all tests pass. The status is set to “failure” if one or more tests fail.
- Script status can be used to conditionally:
 - Alter command flow in repeat/until blocks
 - Alter command flow in if/else/endif commands
 - Conditionally create log entries in the communication view
- There are 26 variables (A to Z). A variable can be used:
 - To alter flow within repeat/until commands
 - As a point index in a control action
 - As an analog output value
 - As a binary output pulse time duration
- Commands in a script file are executed until the script is instructed to stop or until the last command finishes

Commands

Selecting Insert After, Insert Before, or double clicking in an existing script allows creation or modification of a script entry.

Script types are:

Category	Script Name	Description
Communication	Transmit	Send a message
	AddReadObject	Add an object to a read request
Test outstation response	Verify	Verify contents of a response message
	Or	
	And	
Enable/Disable	Enable	Enable or disable DNP3 options
	Disable	
Text Messages	Log	Insert a message in the communication view
	LogIf	
	Prompt	Ask the user a question
Flow Control	If	Control flow based on script success or failure status
	Else	
	Endif	
	Repeat	Repeat a set of script entries until a condition is satisfied
	Until	
Miscellaneous	Restart	Restart from the first script entry
	Stop	Stop script
	Variable	Set or modify the value for one of 26 variables
	Wait	Wait for a specified number of seconds
	ForceError	Cause an error of a certain type

Communication Script Type

- The transmit script type is used to configure any message to be sent.
- The AddReadObject script type is used to add an additional object to a prior "Transmit/Read" type. This is used to, for example, send a request to read both binary and analog inputs in a single message

Transmit configuration options:

Data Link Function

- PRM1_Reset_Link
- PRM1_Request_Link_Status
- PRM1_Unconfirmed_User_Data – Select this whether you plan to use confirmed or unconfirmed user data

Application Function

- Read
- Write
- Select (which automatically sends both select and operate)
- Direct Operate
- Direct Operate (Response and No Response)
- Freeze (Response and No Response)
- Freeze and Clear (Response and No Response)

- Cold Restart
- Enable and disable unsolicited

Other Fields

Object group, variation, qualifier, range, point index, output point value, and CROB information is solicited as appropriate

Verify, OR, And Command Types

These script types set the success or failure status of the script based on the contents of the prior outstation response. All three commands, Verify, Or, And, have the same evaluation criteria. The result of an evaluation is success if all conditions are true. The result is failure if at least one condition is false.

Verify: Script status is set to the result of the command evaluation

Or: Script status is set to success if the command evaluation is success, or unchanged otherwise

And: Script status is set to failure if the command evaluation is failure, or unchanged otherwise

Each script type has several fields. Only those where entries are made are used in the success/failure evaluation.

- Data Link Function, Application Function, Object, Qualifier, and CROB meanings should be obvious. Entry of any field requires the associated response field to contain the value entered.
- Variation is relative to the object entered. A variation cannot be specified if the object field is left blank. If entered, the associated object must be reported with the variation specified.
- Point Index is relative to the object entered. A point index cannot be specified if the object field is left blank. If entered, the defined point and object must be reported
- Point value requires entry of an object type and index. To be successful, the point must be reported with the value entered
- If the **Only if Value Reported in This Message** checkbox is checked, then the point index must be reported in the current response message with the correct value. If this box is not checked, then the test passes if the point is not reported in the current message, but has been reported previously and the value of the most recent report matches the value entered
- If the **Ok if No Response** box is checked, then the test passes if the outstation does not send a response message
- Internal indications can be validated by checking any set of the IINs. As specified, success may be when all the checked IINs are set, when none of are set, or when any of them are set.

Enable/Disable

This script type allows setting or clearing selected DNP3 features. As of this release, the only feature support is User Data Confirm. If not set (default), Unconfirmed User Data is used.

Log, LogIf, and Prompt

These entries allow the coder to enter text messages to present to the user.

Log: unconditionally inserts a message in the communication view

LogIf: inserts a message in the communication view based on the success or failure status of the script. One message can be entered for success and a different message for failure. Either can be left blank if a message is to be logged only on one of the two conditions.

Prompt: specifies a question to ask the user. Script processing continues or aborts based on the answer

If, Else, EndIf

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

Repeat/Until

Repeats a set of script entries until a condition is met. Possible conditions, which may be combined, 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

ForceError

Five error conditions are supported. Each can be:

- Enabled permanently
- Disabled permanently
- Enabled for one message and then automatically disabled
- Generated randomly and occasionally

The five conditions are:

- Send a message with a CRC error
- Do not send an application confirm when requested
- Do not increment the application sequence number
- Ignore outstation requests for time synchronization
- Ignore outstation notifications that it has restarted

Variable

The script editor supports 26 variables, named A through Z. This script entry allows the value of any variable to be set, incremented, or decremented. Variable names can be used as point indices, analog output values, binary output pulse duration, or as a condition to stop or continue a repeat/until loop

Wait/Stop/Restart

- Wait causes script processing to suspend for a given number of seconds
- Stop stops script processing
- Restart restarts script processing at the first entry