



Abstract

The old dEX uses XML format to populate the interface, but it uses text version (which is famous as DXC format). The dEX software will create two files for a program/ job in the 'USER' folder; one is XML format with DEX extension, and the other is text format with DXC extension.

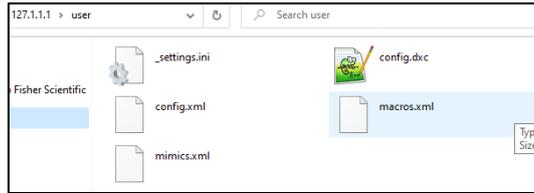


Figure 1: User folder showing config.dxc and config.xml

We remove the dependency of XML format in dEX 2.0 and uses text format only. This text format is quite famous for power users and the actual core of DataTaker programming. The users can see the existence of the original text program in the 'JOBS' folder and jobname subfolder under the name of PROGRAM.dxc.

When you write the program using the old dEX and dEX 2.0, PROGRAM.dxc in the JOBS folder is equivalent to CONFIG.dxc in the USER folder. But if you write the program in freestyle plain text, then PROGRAM.dxc is the representation of your programming style. And you will not find CONFIG.dxc in the USER folder.

Based on the similarity of CONFIG.dxc between the old dEX and dEX 2.0, this note will explain the conversion process for making the old dEX dxc version compatible with dEX 2.0. We cover the discussion about freestyle text conversion other notes.

Introduction

Although most dxc components between the old DEX and dEX 2.0 are the same, a few are different, especially manual and alarm channel declaration. We add a few modifications to the programming style related to alarm declaration (now can be declared with multiple conditions), and some specific channels that the users need to declare it under a manual channel (i.e., MODBUS).

If the program (written in the old dEX) already in the logger, dEX 2.0 can convert it straight away if there is no manual channels (i.e., MODBUS channels) or alarms declaration. However, if you have manual channels or alarm declaration, you will find a "dxc compilation error" when attempting the conversion.

If users have stored a dex file in their computer rather than running it in the logger, this note will assist them with the conversion.

DXC Extraction from the old dEX Program

1. If the old program is already in the logger, you can use DeTransfer and send SHOWPROG to get the program. Copy and paste it to a notepad and give any name with the dxc extension.

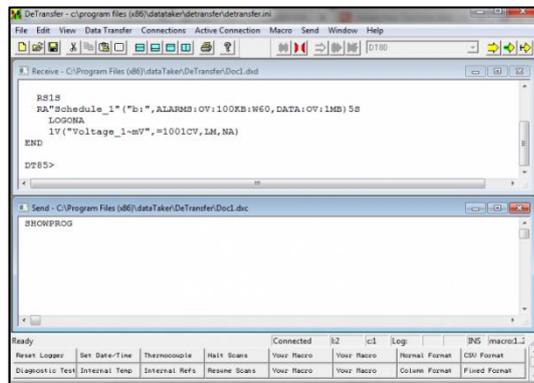


Figure 2: SHOWPROG in DeTransfer

2. If the old program is already in the logger, you can access the dataTaker FTP server and go to the 'USER' folder and extract config.dxc file, check figure 1.
3. If you have a stored dex file in the computer and you want to use it on dEX 2.0, you have to extract the program as a text. Unfortunately, the only solution would be dEX 1.0 (or commonly known as dEX for desktop). I do not recommend using dEX 1.0 for prolong time other than extracting the text version of the program due to a change of ownership from Adobe to Harman.

This third option requires installing Adobe Air (I put 2 version since I am not sure which one will work, try version 25 first) and dEX for desktop application. You can download Adobe Air and dEX 1.0 from the following link:

- Air Version 25: <https://datatakerforum.com/Documents/Applications/AdobeAIRInstaller.exe>
- Air Version 32: <https://datatakerforum.com/Documents/Applications/Adobe-Air-32-0-0-125.exe>
- dEX 1.0: https://datatakerforum.com/Documents/Applications/dex_416_code1_desktop.air

Modification

This is one example of text program extracted from the old dEX, you will need to use notepad or notepad++ in order to modify the text:

```

BEGIN"config"
' Generated by dEX Configuration Builder Version 1.85.002 (Firmware Version 9.22.9125)
' Target device: DT85M3-4
password=""

RS1S
RA"Schedule_1"("b:",ALARMS:OV:100KB:W60,DATA:OV:1MB)5M
LOGONA
ALARM1(1V("Voltage_1~mV",=1001CV,LM,NA)<5)"Event triggered on Logger ! at @ on #, ?N value is ?V ?U."{}

' User defined
1MODBUS(AD1,R4:1001,MBF);

END

```



a. Alarm section

In dEX 2.0, the alarm is no longer part of a channel, but it is now a separate channel; therefore, you can have multiple conditions at once. This new feature is the complete opposite of the previous dEX.

```
ALARM1(1V("Voltage_1~mV",=1001CV,LM,NA)<5)"Event triggered on Logger ! at @ on #, ?N value is ?V ?U."{}
```

In the old dEX, as you can see in the above example, the channel was becoming a channel option for alarm. The alarm was taking the primary role. You can modify the above alarm by splitting channel and alarm, grab the entire channel `1V("Voltage_1~mV",=1001CV,LM,NA)` and put it as a new line above the alarm channel, then in the original location, replace the channel with a CV which is `1001CV`.

```
1V("Voltage_1~mV",=1001CV,LM,NA)
ALARM1(1001CV<5)"Event-triggered on Logger ! at @ on #, ?N value is ?V ?U."{}
```

b. Manual channel section

You can modify the manual channel as follows, make sure there is no space/ tab before each line within 'Manual declaration (start and end syntax + content). There are **four lines limit** for each manual channel; thus, if you initially have long channels, you must split them into several parts in dEX 2.0.

Original syntax:

```
'User defined
1MODBUS(AD1,R4:1001,MBF);
```

Modified syntax:

```
'Manual "Manual"{
1MODBUS(AD1,R4:1001,MBF)
'Manual}
```

After addressing those two components, the final modified program will look like the following:

```
BEGIN"config"
' Generated by dEX Configuration Builder Version 1.85.002 (Firmware Version 9.22.9125)
' Target device: DT85M3-4
password=""

RS1S
RA"Schedule_1"("b:",ALARMS:OV:100KB:W60,DATA:OV:1MB)5M
LOGONA

1V("Voltage_1~mV",=1001CV,LM,NA)
ALARM1(1001CV<5)"Event triggered on Logger ! at @ on #, ?N value is ?V ?U."{

'Manual "Manual"{
1MODBUS(AD1,R4:1001,MBF)
'Manual}

END
```



You must save that script into a dxc file, and you can upload it into dEX 2.0. Click the three dots next to the “Apply to Logger” button and choose “Load from a File.”

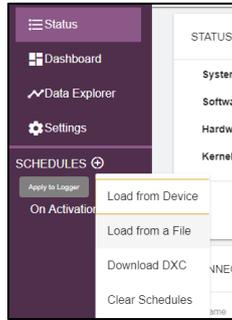


Figure 3: Loading from dxc file