Anonymous Measurement in CANape

2020-06-29
Support Note SN-IMC-1-008_Anonymous_Measurement_in_CANape

Author(s) Reusch, Axel; Wittmer, Helena; Holl, Jonathan
Restrictions Public Document

Table of Contents

1 Overview ... 2
1.1 Anonymous Measurement ... 2
1.2 Roles and rough steps ... 3
1.2.1 Role “Creator” | Create CNAX file ... 3
1.2.2 Role “Creator” | Adjust A2L File ... 4
1.2.3 Role “Test Driver” | Use CNAX file ... 6
1.2.4 Role “Offline Analyst” | Analyze MDF created with CNAX ... 7
1.3 Deanonymization via CallConverter ... 8
2 Contacts ... 9

1 Overview

This Support Note describes the required steps to create, use and analyze a configuration in CANape in the context of “Anonymous Measurement”.

Note
Screenshots in this document taken from CANape 18.0

1.1 Anonymous Measurement

Imagine you are a development engineer and you are testing a new component in a vehicle and

- you want to measure a lot of data coming from the ECU during test drives
- you and your colleagues can’t achieve the goal of collecting that much data due to missing resources
- the signals you want to measure have to be treated confidential
- you want to enable some development partners outside your company to measure the ECU data for you.

How can this be achieved?

The main benefit of “Anonymous Measurement” is the possibility to handover measurement tasks to development partners without revealing the meaning of the recorded data.

For “Anonymous Measurement” the configuration does not visualize signal names - neither in the CANape GUI nor in the created `MDF` file. The acquisition of measurement data and calibration objects is possible without signals/objects contained within the `A2L` database.

The extension of the corresponding configuration file is `CNAX` (instead of usual CANape configuration: `CNA`).

“Anonymous Measurement” can be used for any CANape device based on an `A2L` file (XCP, CCP). “Anonymous Measurement” is not relevant for measuring bus messages (i.e. CAN Monitoring) as this can be achieved without special CANape functionality by using a bus recording database file (i.e. `DBC`) without relevant messages (measure complete raw data of bus).

Please regard:
A `CNAX` file should be used to allow measurement of signals that usually should not be displayed for specific CANape users. Therefore it is not intended to, i.e. visualize complex signals in detail (like map objects) or perform calculation of functions during measurement – this might lead to warning messages when opening the `CNAX`. Further visualisation or analysis of signals is executed during analysis phase, based on the recorded `MDF` file and the original `A2L` file.

Therefore: Please create a minimized configuration file especially for the use case of measuring relevant data, but not necessarily visualizing it. This procedure results in a reasonable `CNAX` file.

Note
The feature “Anonymous Measurement” is patented within the EU.

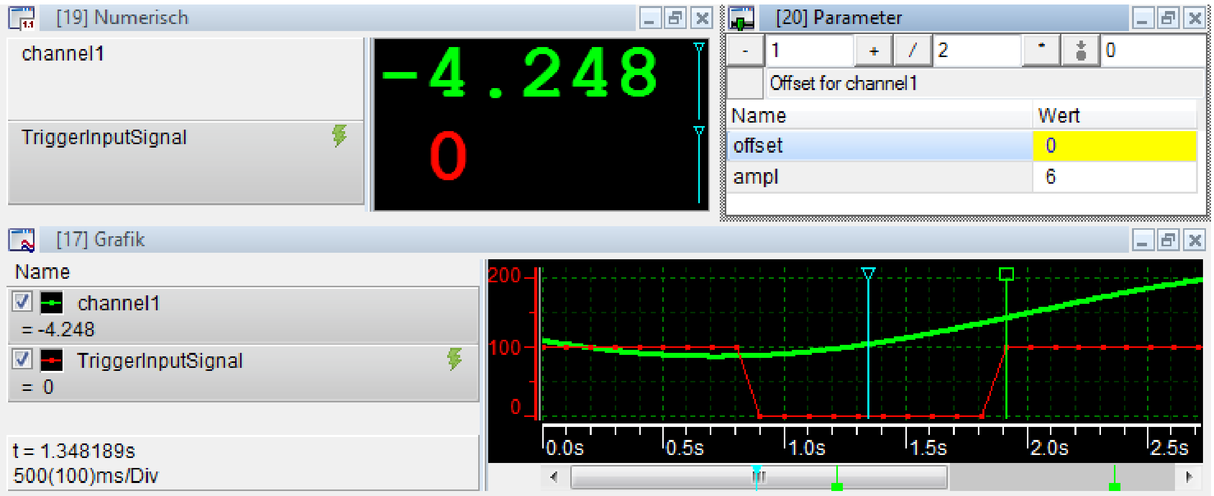
Step by Step

This Support Note will lead you through the required steps of the corresponding roles. (See following chapters)

1.2 Roles and rough steps

1.2.1 Role “Creator” | Create CNAX file

The “Creator” generates a `CNAX` file based on an especially created `CNA` file of a working project. The required signals should be contained within the measurement configuration to record any relevant data. Some signals might be contained within measurement/calibration windows, i.e. like this (some windows of modified XCP example project):



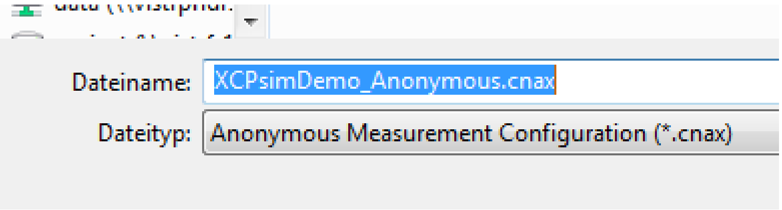
Name	Wert
offset	0
ampl	6

If triggers are used within the configuration the functionality will persist as expected when using a `CNAX`.

Note

Though the name is not visible, the address and length of a signal can be investigated when using “Anonymous Measurement”. If this should also be prevented, measuring structures, value blocks or arrays which contain the relevant signal data is a possible solution. That way a larger address block is measured and afterwards one or several particular signals (address and length) contained within that block can be analyzed.

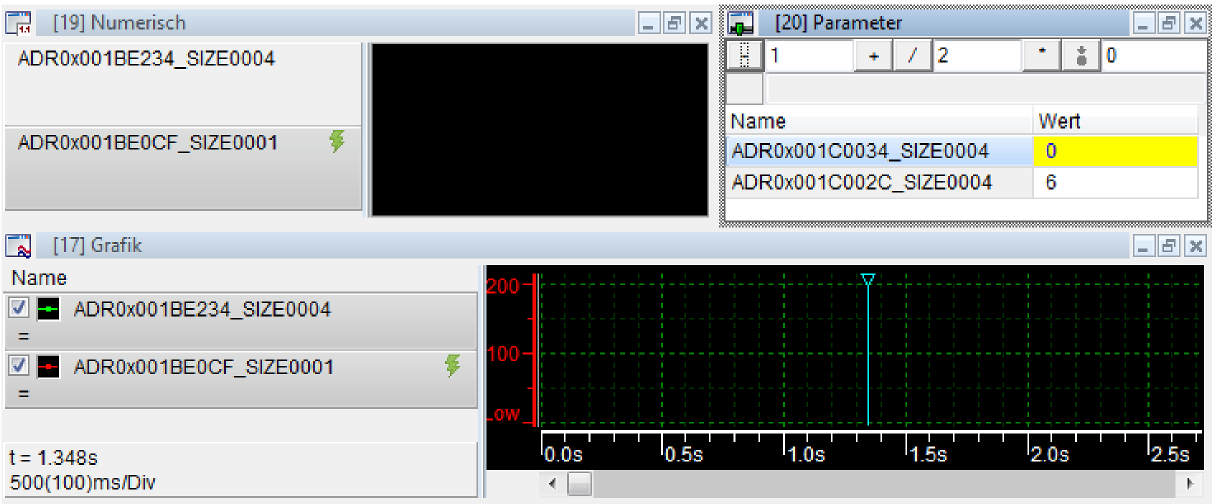
Using this valid `CNA` as starting point, a `CNAX` file can be created by selecting **Configuration|Save As|*.cnax** in CANape backstage area.



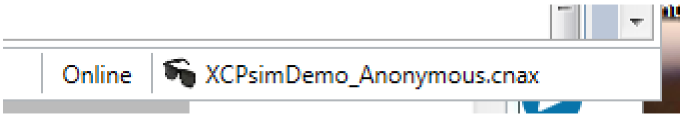
Dateiname: XCPsimDemo_Anonymous.cnax

Dateityp: Anonymous Measurement Configuration (*.cnax)

After saving and reloading the CNAX file no signal names are contained in the same windows (and measurement configuration, Symbol Explorer…) anymore:



In addition, the status bar shows a special icon next to the CNAX file name to visualize that a CNAX file is in use:

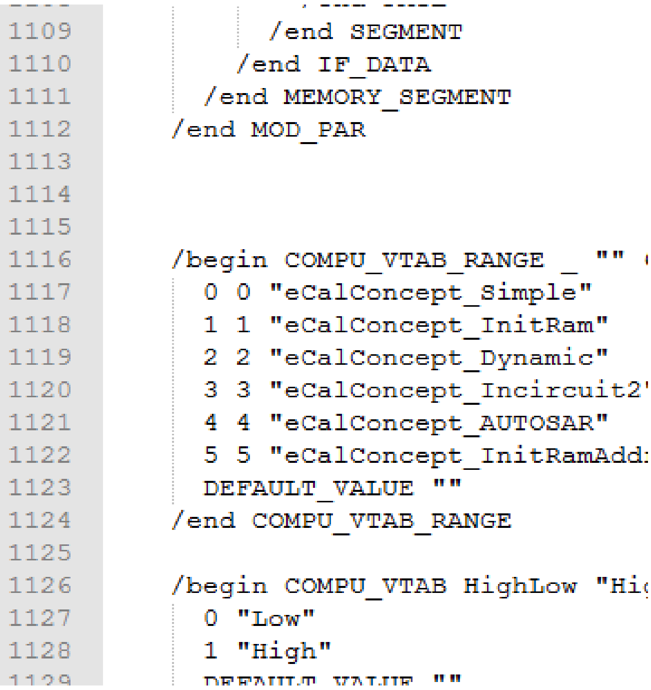


1.2.2 Role “Creator” | Adjust A2L File

Now, when opening the CNAX file any signal name is disclosed within the CANape GUI. However, at this point it would still be possible to investigate the A2L file to find out what signals are used. The relevant signals should not be contained within the A2L file!

Working with a CNAX does not require signal definitions within the A2L file. For measurement and calibration with CANape the CNAX file can be delivered with a reduced A2L file, without any signal definition contained.

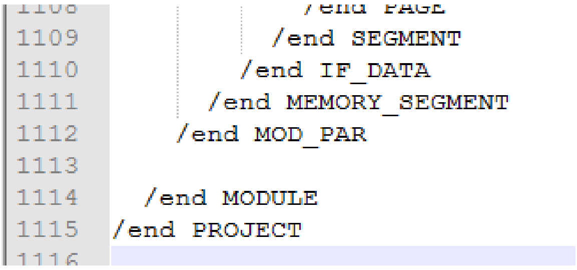
Please regard: Nevertheless, it is necessary to provide an A2L file with required driver data to ensure correct communication with the ECU – working without an A2L file is not possible.
Here is an excerpt of the XCPSim.a2l of CANape example XCP project before the modification, the file still contains relevant signal data below the required driver data (end with /end MOD_PAR):



```
1108          /end PAGE
1109        /end SEGMENT
1110      /end IF_DATA
1111    /end MEMORY_SEGMENT
1112  /end MOD_PAR
1113
1114
1115
1116  /begin COMPU_VTAB_RANGE _ "" (
1117    0 0 "eCalConcept_Simple"
1118    1 1 "eCalConcept_InitRam"
1119    2 2 "eCalConcept_Dynamic"
1120    3 3 "eCalConcept_Incircuit2'
1121    4 4 "eCalConcept_AUTOSAR"
1122    5 5 "eCalConcept_InitRamAdd:
1123    DEFAULT_VALUE ""
1124  /end COMPU_VTAB_RANGE
1125
1126  /begin COMPU_VTAB HighLow "Hi(
1127    0 "Low"
1128    1 "High"
1129    DEFAULT_VALUE ""
```

Various ways will lead to a reduced `A2L`:
An easy possibility is the export of an `A2L` file from a CNA file with signals that are allowed to be contained within the target `A2L` via menu **Tools | Export | Configuration** (some signals are contained in `A2L` afterwards). This procedure can also be used if you are using an empty `CNA` file without signals contained in GUI and measurement configuration (no signals will be contained in target `A2L`). A further possibility is the modification of an `A2L` file via ASAP2 Studio. Other tools might be used as well for that purpose.

After the modification, the file excerpt should look like this (end of `A2L` file is at line 1115). There is no signal relevant information contained anymore:



```
1108          /end PAGE
1109        /end SEGMENT
1110      /end IF_DATA
1111    /end MEMORY_SEGMENT
1112  /end MOD_PAR
1113
1114  /end MODULE
1115 /end PROJECT
1116
```

Please doublecheck that no relevant information is contained within the `A2L` file (customer name, remains of signal information like record layouts…), i.e. via text editor.

As it is possible to create an `A2L` file with a subset of signals an usual `CNA` file can be used together with the `CNAX` file to cover both use cases (Multiple Configuration). The standard `CNA` can be used to measure and visualize the signals as usual, the additionally available `CNAX` can be used for “Anonymous Measurement”.

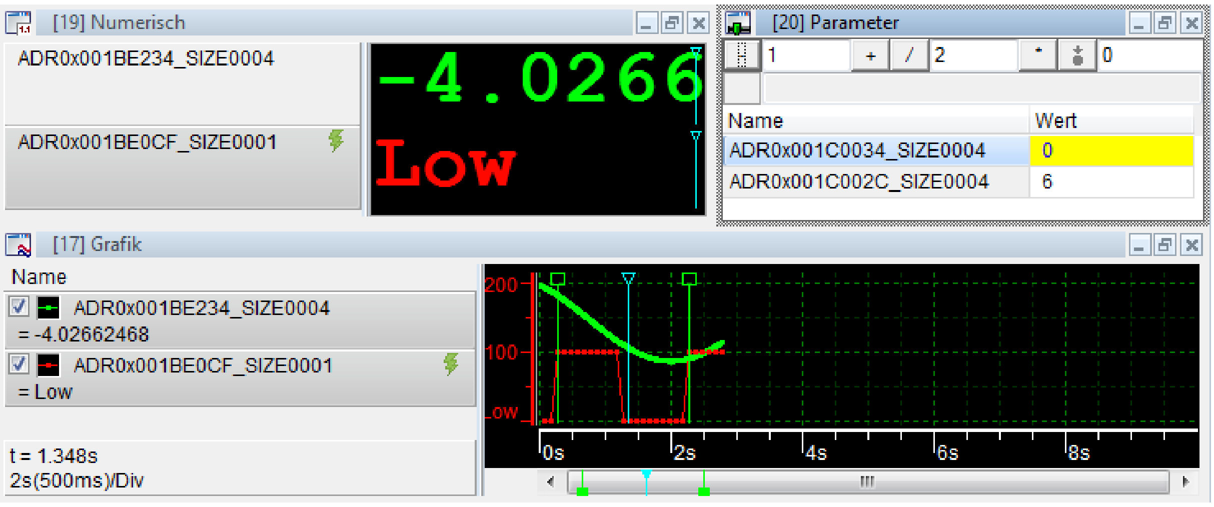
1.2.3 Role “Test Driver” | Use CNAX file

The mandatory files to work with “Anonymous Measurement” are:

- CNAX file
- A2L File (reduced, no/reduced signal relevant definitions)

By using the A2L file it is now possible to create a device within a new CANape project. As the A2L provides no signal definition it is not possible to add any signal to the configuration for measurement purposes.

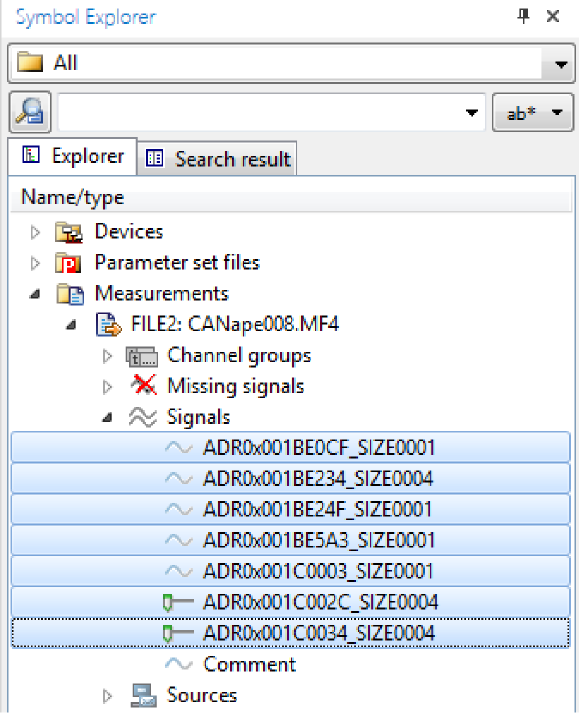
However, after loading the CNAX file (backstage area of CANape, **Configuration | Load | *.cnax**), it is possible to measure the specified signals anonymously.



Name	Wert
ADR0x001C0034_SIZE0004	0
ADR0x001C002C_SIZE0004	6

It is not possible to analyze a saved MDF file in a valid way without the corresponding A2L file. However, any saved MDF file can be used for later analysis with the correct, original A2L file.

Here is an example of an MDF recorded with a CNAX without a corresponding A2L file:

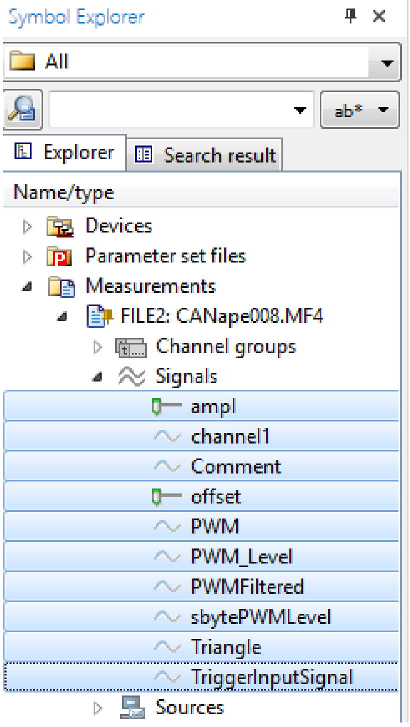


- Devices
- Parameter set files
- Measurements
  - FILE2: CANape008.MF4
    - Channel groups
    - Missing signals
    - Signals
      - ADR0x001BE0CF_SIZE0001
      - ADR0x001BE234_SIZE0004
      - ADR0x001BE24F_SIZE0001
      - ADR0x001BE5A3_SIZE0001
      - ADR0x001C0003_SIZE0001
      - ADR0x001C002C_SIZE0004
      - ADR0x001C0034_SIZE0004
      - Comment
    - Sources

1.2.4 Role “Offline Analyst” | Analyze MDF created with CNAX

As any `MDF` file created with a `CNAX` does not contain signal definitions, the corresponding `A2L` with signal definition is required for analysis.

Within a project with the original `A2L` the recorded `MDF` file can be loaded as usual and will be interpreted, like this:

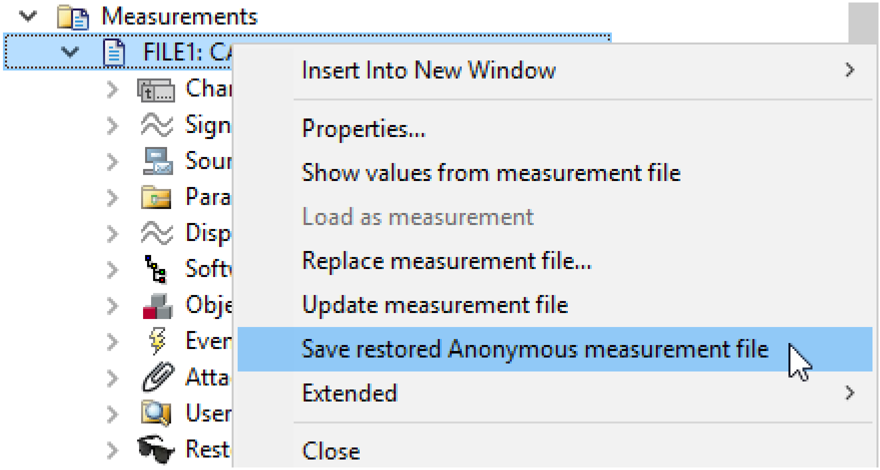


Required: The current project must contain a respective device with

- correct device name
- correct name of `A2L` file
- correct path to `A2L` file (only until CANape 16.0 SP4)

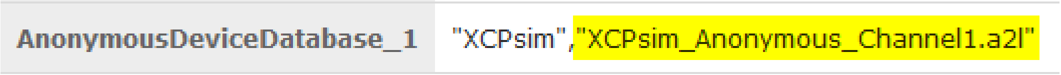
The restored signal description is not automatically saved into the loaded measurement file. This may be executed by the user via corresponding menu command in the Symbol Explorer or by calling script method `RestoreMeasurementFile()`.

In CANape 18.0 it’s possible to convert `MDF3` and `MDF4` files. Please regard that in earlier CANape Versions it’s only possible to convert `MDF3` files, `MDF4` files can only be analyzed in the original CANape project.



1.3 Deanonymization via CallConverter

Starting with the release of CANape 18.0 it will be possible to deanonymize the recorded file via command line outside of CANape.
The required file aside from the measurement files is the original `A2L` file which has the same name as the `A2L` used for the measurements, it can be seen in the Measurement file Comment:

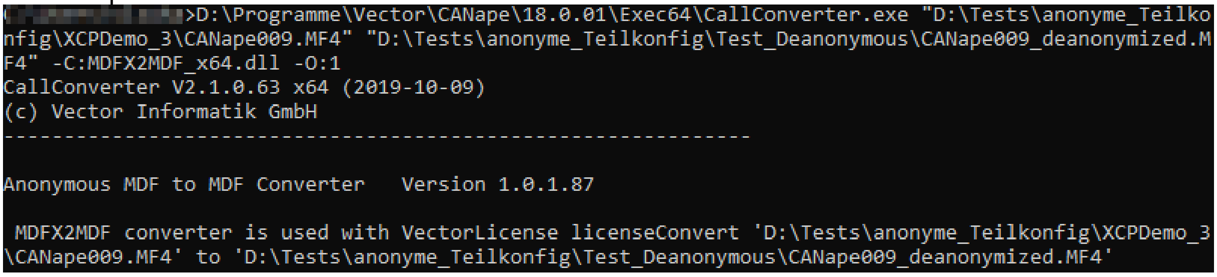


AnonymousDeviceDatabase_1	"XCPsim","XCPsim_Anonymous_Channel1.a2l"

The `A2L` file has to be located in the same folder as the measurements or else the CallConverter won’t be able to find it. The name of the `A2L` is defined by the measurement file comment and doesn’t need to be defined in the command line.
The command line for the CallConverter should contain the following Information:

```
[Path to CANape Folder “Exec64”]\CallConverter.exe “[Path to anonymous
measurement file]\Measurement.MF4” “[Destination
folder]\deanonymized_Meas.MF4” -C:[Path if different from
CallConverter.exe]\ MDFX2MDF_x64.dll -O:[Logfile Information]
```

An example would look like this:



```
[redacted]>D:\Programme\Vector\CANape\18.0.01\Exec64\CallConverter.exe "D:\Tests\anonyme_Teilko
nfig\XCPDemo_3\CANape009.MF4" "D:\Tests\anonyme_Teilkonfig\Test_Deanonymous\CANape009_deanonymized.M
F4" -C:MDFX2MDF_x64.dll -O:1
CallConverter V2.1.0.63 x64 (2019-10-09)
(c) Vector Informatik GmbH
--------------------------------------------------------------

Anonymous MDF to MDF Converter   Version 1.0.1.87

 MDFX2MDF converter is used with VectorLicense licenseConvert 'D:\Tests\anonyme_Teilkonfig\XCPDemo_3
\CANape009.MF4' to 'D:\Tests\anonyme_Teilkonfig\Test_Deanonymous\CANape009_deanonymized.MF4'
```

If -O is not used, the target file won’t be overwritten, and the execution will be cancelled without an error.
If no name for the output `MDF` is defined, a new name will be generated automatically `(“*_deanonymized”).`
`CallConverter.exe` is normally located in the `Exec64` folder in CANape, the full path has to be used if the command line is called from another folder.

2 Contacts

For support related questions please address to the support contact for your country https://www.vector.com/int/en/company/contacts/support-contact/.