

Automated Flashing and Testing with CANoe, vFlash and VN89xx

Version 1.0

2019-02-19

Application Note AN-IDG-1-018

Author Thomas Schmidt

Restrictions Public Document

Abstract Automatic updating an ECU's software with vFlash and running tests with CANoe

Table of Contents

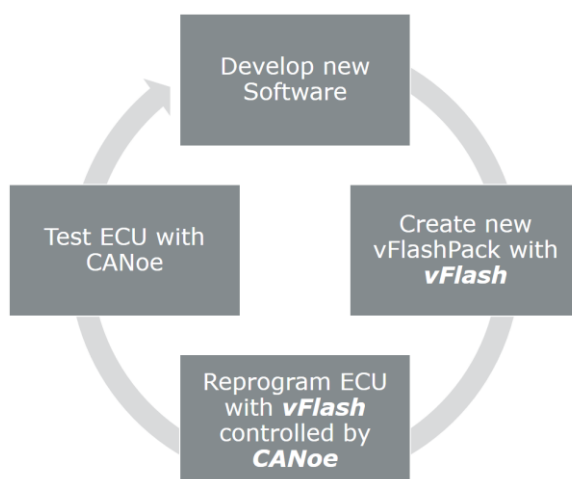
1	Overview	2
1.1	Concept	2
1.2	Environment.....	2
2	Flashing and Testing	3
2.1	Creation of vFlashPack	3
2.2	Reprogramming ECU.....	3
2.3	Run CANoe tests	5
3	Automation	6
4	Appendix	7
4.1	Function to copy vFlashPack "user file" to vFlash project folder	7
4.2	Example: VisualBasicScript to create vFlashPack and start CANoe test	8
5	Contacts	9

1 Overview

During ECU development ECUs must be updated frequently with new software. Afterwards the new software needs to be tested – ideally with automated regression tests. CANoe and vFlash offer a perfect environment to automate the process of updating and testing.

1.1 Concept

This application note offers a proposal on how to use CANoe, vFlash and the modular network interface system VN89xx or VT System to automate creating a packed vFlash project file (.vFlashPack¹) with vFlash, downloading the software with vFlash and executing regression tests with CANoe.



1.2 Environment

To create a vFlashPack vFlash needs to be installed on a PC. To run the CANoe tests including reprogramming CANoe and vFlash needs to be installed on the VN89xx.



Note

An active CANoe and vFlash license is required on the VN89xx. The vFlash license needs to be located on the VN89xx itself and not on an additionally connected network interface (e.g. VN5610).

¹ vFlashPack = vFlash project that contains all relevant project information in one single file – including flashware files. More detailed information is available in the vFlash Help.

2 Flashing and Testing

2.1 Creation of vFlashPack

Whenever the ECU software has been modified and a new package is built several updated flashware files are created. These flashware files are either in binary format, hex format or even proprietary format.

Pre-condition

To create a vFlashPack a non-packed vFlash project (.vFlash) must be available, that references all flashware files including checksum files, signature files and so on. Whenever the new software is built the referenced files must be updated. Based on the existing non-packed vFlash project the new vFlashPack can be created.

Creation

To create a new vFlashpack you can either use vFlash via command line or via graphical user interface (GUI):

1. Command line:

```
vFlash /ExportToPack D:\Data\Project\Project.vflash ↵  
D:\Data\CANoeConfig\Project.vflashpack
```

2. GUI:

- > Open the vFlash project file in vFlash
- > Execute "Export to Pack&Go" via menu [File | Export to Pack&Go]

Alternatively, if flashware files have not been overwritten ...

- > Reconfigure the vFlash project by selecting the new flashware files
- > Execute "Export to Pack&Go" via menu [File | Export to Pack&Go]

2.2 Reprogramming ECU

CANoe can control vFlash to reprogram an ECU using the new vFlashpack. A special CANoe configuration is required that uses the vFlashNodeLayer.DLL to control vFlash.

Initially the new vFlashPack must be copied onto the VN89xx. Afterwards CANoe can run the test module that reprograms the ECU before it runs the regular ECU regression test modules.

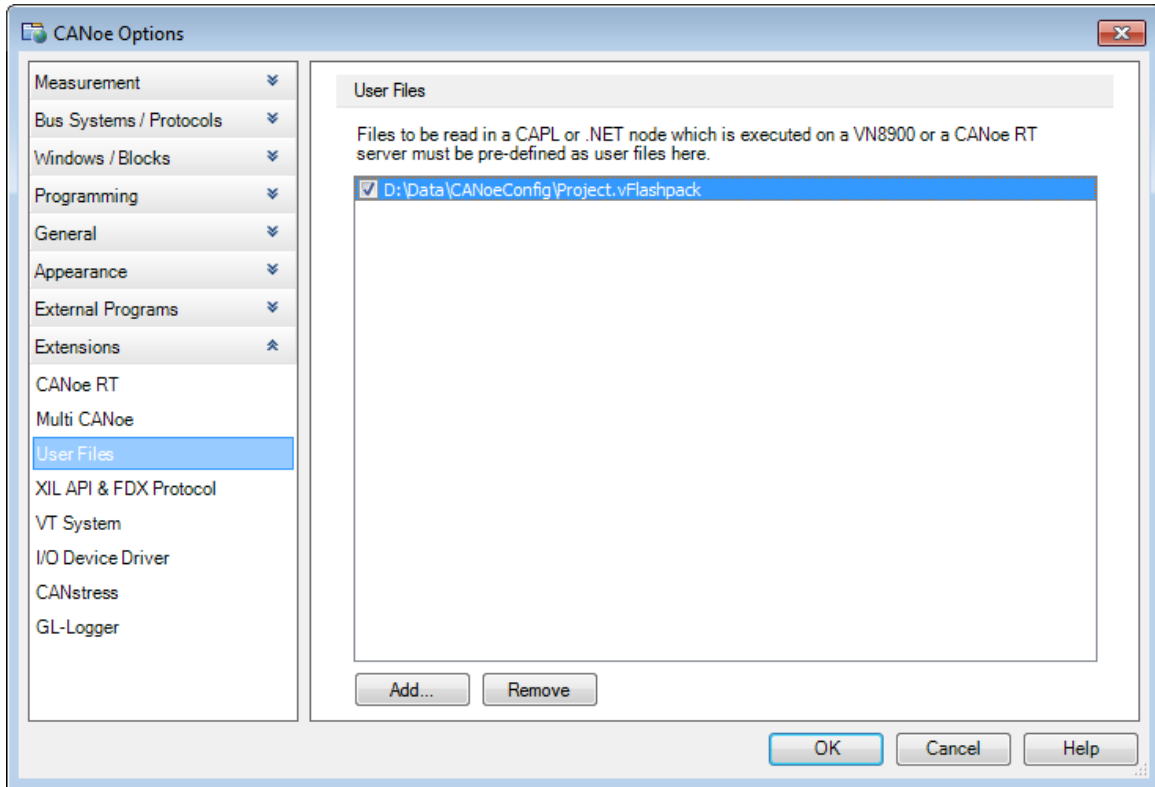
Copy vFlashPack to VN89xx

- ▶ The CANoe configuration must reference the vFlashpack as "User File" to ensure that the file is transferred to the RT system by CANoe automatically.



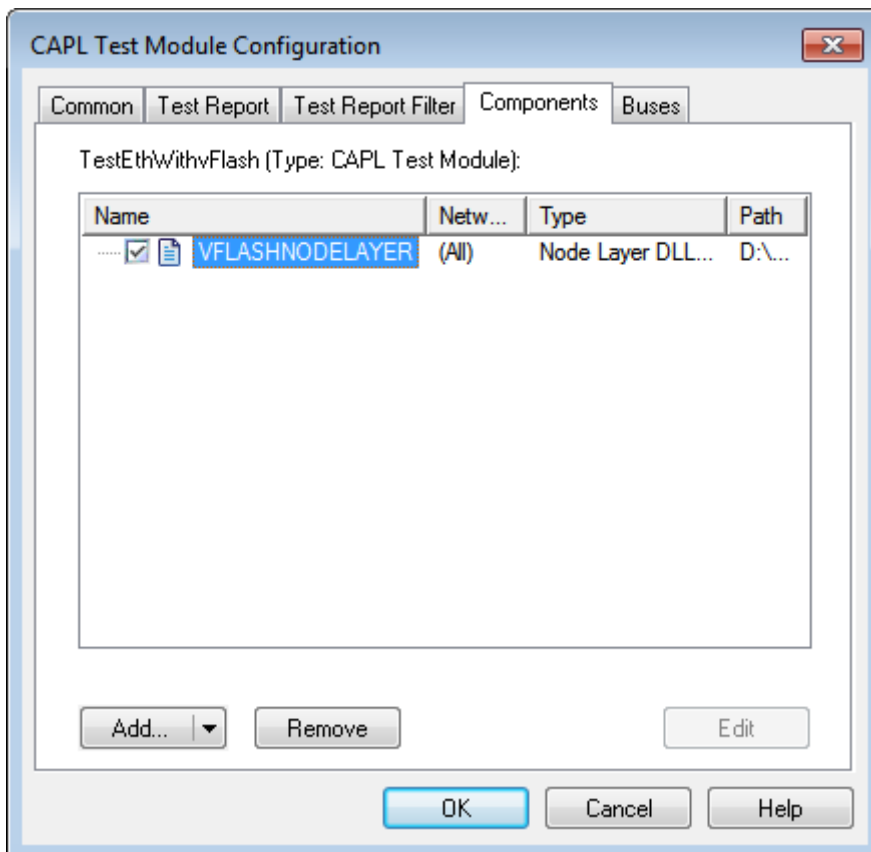
Note

Before CANoe 12.0 the vFlashPack is copied to the VN89xx into a folder not accessible for vFlash. Therefore, the CANoe test module must copy the vFlashPack into the vFlash project folder on the VN89xx (see the Appendix to get a complete implementation).



Reprogram the ECU

- > A CANoe CAPL test module can use the vFlashNodelayer.DLL to control vFlash on the VN89xx. In the configuration of the test module the DLL must be added as “component”:



- ▶ CANoe provides an implementation of a test command `TestWaitForvFlashPackReprogrammed` that flashes one vFlashpack in the include file "vFlash/Utilities.cin". It can be used as follows:

```
/* CAPL TEST MODULE: */
includes
{
    #include "vFlash/Utilities.cin" // installed with CANoe
}

variables
{
    // This is the user file that is automatically synchronized.
    // vFlash will search it under a specific path, therefore it must be copied.
    char gFlashpack[100] = "Project.vflashpack";
}

Testcase long TC_FlashPack(char vFlashPackName[])
{
    #if TOOL_MAJOR_VERSION < 12
        // vFlash expects the pack at a specific location, so copy it.
        // See appendix for a possible implementation
        CopyvFlashpackTovFlashFolder(vFlashPackName);
    #endif // CANoe before 12.0

    strncpy(gFlashpack, vFlashPackName, elcount(gFlashpack)); // No path necessary
    return TestWaitForvFlashPackReprogrammed(""); // argument is ignored
}
```

2.3 Run CANoe tests

After the new ECU software has been successfully programmed into the ECU, the application tests can be executed. A complete CAPL test module may look like following:

```
/* CAPL TEST MODULE: */
MainTest()
{
    // Flash ECU - no path necessary since it is a CANoe "user file"
    if (Success != TC_FlashPack("ECU1.vflashpack"))
    {
        return; // do not continue if flashing the new software failed
    }
    TC_Application();
}

Testcase TC_Application()
{
    ...
}
```

3 Automation

Using the instructions of chapter 2, it is possible to sequentially Flash and Test the new application software. This can also be fully automated.

- > Create a vFlashpack from the vFlash project using a command line switch “/ExportToPack” (note that absolute paths must be used). This can be done in a simple Windows batch file, e.g.
- > Start CANoe (which must be connected to the VN89xx on which CANoe RT runs), load the configuration and start the measurement using the COM API, e.g. from .NET. You will find a trivial VisualBasicScript as an example in the appendix.

4 Appendix

4.1 Function to copy vFlashPack “user file” to vFlash project folder

Until CANoe 12.0 it is necessary to copy the synchronized vFlashPack from the user files folder into the vFlash projects folder on the remote device. This can be achieved by calling the Windows program `xcopy` which is available on all installations.

Note that the files must be copied to the remote device first, and that this trivial implementation assumes that copying the file does not need more than 500 ms.

```
// Copy a vFlashpack that has been configured as "user file" for vFlash to find
// Return 0 if there was a problem retrieving the file
int CopyvFlashpackTovFlashFolder(char flashPackName[])
{
    char synchronizedFile[200];
    char commandLine[200];
    long status;

    // Make sure the file's content is present on the RT side
    status = testWaitForUserFileSync(flashPackName, 1); // sync to RT side
    if (1 != status)
    {
        TestStepFail("Copy", "Cannot synchronize user file '%s' to RT side! error %d",
                    flashPackName, status);
        return 0;
    }

    // Find its actual path
    if (0 >= getUserFilePath(flashPackName, synchronizedFile,
                            elcount(synchronizedFile)))
    {
        TestStepFail("Copy", "Error retrieving synchronized path of flashpack '%s'!",
                    flashPackName);
        return 0;
    }

    // Copy the file synchronized onto the VN89xx to where vFlash expects it
    sprintf(commandLine, elcount(commandLine), "/Y \"%s\" D:\\vFlashProjectFiles\\",
            synchronizedFile);
    TestStep("Copy", "Executing command line:\n%s", commandLine);
    sysExecCmd("xcopy.exe", commandLine);

    // We do not have a simple way to check if the file has been copied,
    // so just wait some time...
    TestWaitForTimeout(500);
    return 1;
}
```

4.2 Example: VisualBasicScript to create vFlashPack and start CANoe test

This VBScript example shows how vFlash can be used to create a vFlashPack from a project. It includes an updated code file (.bin/...) into the .vflashpack file. CANoe copies the vFlashPack to the remote machine automatically. You can start the measurement from the script, too.

```
#####
'# Example implementation: Create an updated vFlashPack, then load
'#                               and start a CANoe(RT) test configuration
'# (c) Vector Informatik
#####

Option Explicit
On Error Resume Next
Dim App, objShell, pathvFlashExe, pathvFlashProject
Dim cfgPath, loaded
'
' !!! Update the paths used here !!!
'
pathvFlashExe = ""D:\VectorTools\Vector vFlash 4\Bin\vFlash.exe""
pathvFlashProject = "D:\Support\2018-12-19_EthOnVNwithUpdateFlashpack\"
cfgPath = "D:\Support\2018-12-19_EthOnVNwithUpdateFlashpack\EthWithFlashing.cfg"

' Create an updated flashpack from the vFlash project.
' Note: the project must reference the updated .bin/... file!
Set objShell = CreateObject("WScript.Shell")
objShell.Run pathvFlashExe & " /ExportToPack "" & pathvFlashProject & _
    "ControlCANoeCOM\ECU1.vflash"" "" & pathvFlashProject & "ECU1.vflashpack"", _
    6, True
Set objShell = Nothing

' Make sure CANoe is running
Set App = CreateObject("canoe.Application")
Err.clear

' "Connect" callbacks to app to get a call when the configuration was loaded
WScript.ConnectObject App, "App_"
If Err.number <> 0 Then
    WScript.echo "ConnectObject app failed: 0x" & Hex( Err.number)
    Return
Else
    ' WScript.echo "ConnectObject app worked."
End If

' Load the configuration and wait until that is done
loaded=0
App.Open cfgPath, False, False
While 0 = loaded
    Wscript.Sleep(10)
    ' Wscript.echo "Not loaded yet"
Wend

' Simply start the measurement
App.Measurement.Start

' This is called by CANoe when the configuration is loaded
Sub App_OnOpen(fullname)
    ' Wscript.echo "Loaded."
    loaded = 1
End Sub
```


5 Contacts

For a full list with all Vector locations and addresses worldwide, please visit <http://vector.com/contact/>.