XMLCLI Setup and guide to access the UEFI Variables

**Note:** *Refer the ReadMe.txt (at <Path\_To\_XmlCliRefScripts>) and it will help you to configure XmlCli, below will help to configure the XmlCli only in Target windows SUT*

# Configuring Xml in Target Windows SUT

*Note: You can find the required software in below share path.* [\\bascrd101\svshare\Amol\XmlCliRefScripts](file:///\\bascrd101\svshare\Amol\XmlCliRefScripts)

1. Install the python 2.7.X version or later (python-2.7.6.amd64.msi)
2. Install the CCB SDK software (Intel® CCG Platform Tools SDK.zip)
3. Make sure that supported BIOS version is already flashed on the Target board, otherwise please contact the BIOS team for right BIOS version to use this.
4. Copy the Reference scripts(xmlcli\_Rev0.97.zip) on any folder on the system
5. Open Python Prompt with administrator privileges and run following commands to do common importing

*import sys*

*sys.path.append(r"<Path\_To\_XmlCliRefScripts>")*

*import XmlCli as cli*

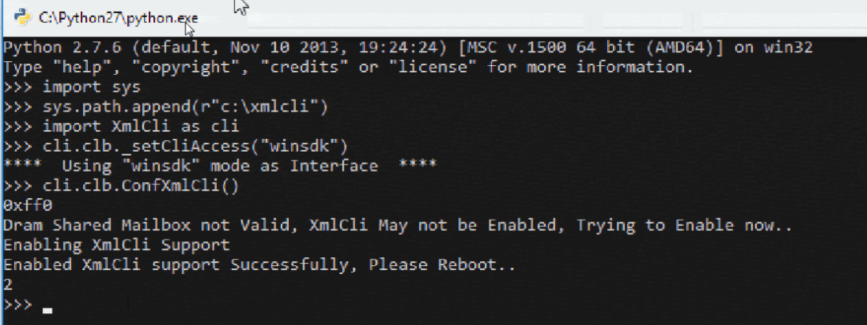
1. Run below command in python console to setting the SDK interface to run XmlCli in Windows SUT

*cli.clb.\_setCliAccess("winsdk")*

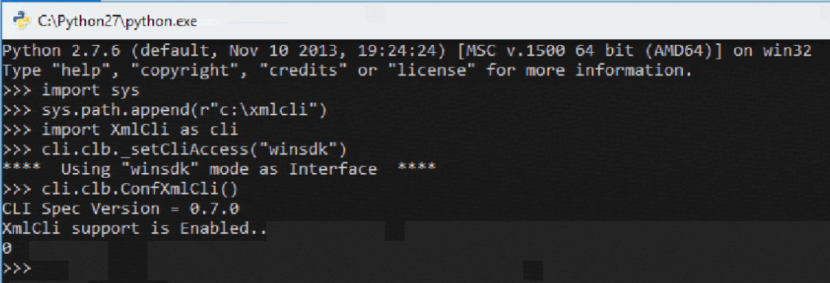
1. Run the below command to enable or check the status of XmlCli support flag

*cli.clb.ConfXmlCli()*

* If status is return as 2, then reboot the system to load XmlCli drivers and enabling the support.

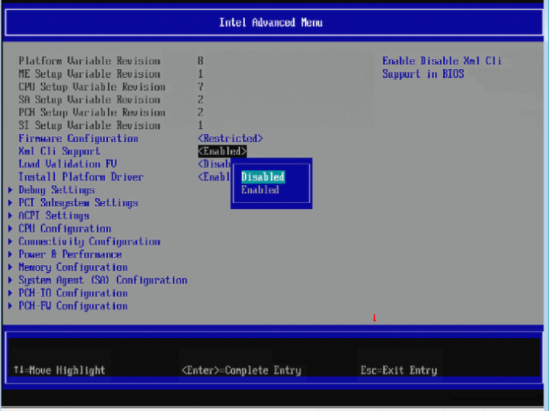


* If status is return as 0 , then XmlCli support has been enabled properly



*Note: You can enable XmlCli from BIOS menu options as below.*

**“BIOS menu - > Intel Advanced Menu -> Xml Cli Support -> Enabled”**



# UEFI Get/Set/Gms Command

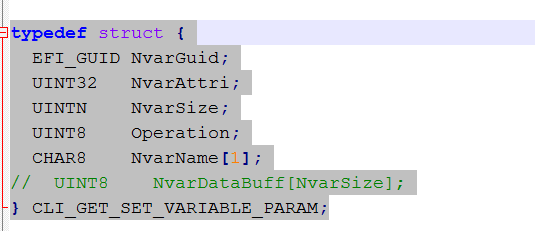
cli.GetSetVar ( ‘VarOperation’,VarMode, ‘VarMemberSet’,”VarName”,”VarGuid”,”VarAttributes”,”VarSize”,’VarDataBytes’)

|  |  |
| --- | --- |
| **Arguments** | **Description** |
| VarOperation | String indicates to action on variable and defined operation are as below.   * ‘get’ – Read the existing UEFI Variables (GetVariable) * ‘set’ – Write the data into exiting UEFI Variables (SetVariable) * ‘gms’ –Modify the UEFI variables data (GetVariable/Setvariable) |
| VarMode | String indicates to mode of command that whether inputs are in form of Command Line arguments or via XML file.   * 0 – Indicates that inputs are to be via Command Line arguments * r”path/filename.xml” - Indicates that inputs are to be via given XML file |
| VarMemberSet | String which contains Variable members to be modify with given value. This is used for ‘gms’ operation   * ‘’ – this to be used for ‘get’ and ‘set’ operation * ‘VarName.First4BytesOfVarGuid.MemberName=Value’ – This to be used for ‘gms’ opeartion |
| VarName | String which contains UEFI Variable Name - Null terminated Unicode string |
| VarGuid | String which contains Variable GUID – 16 byte unique identifier |
| VarAttributes | String which contains Attributes to the UEFI variable, usually “0x07” will be used – 4 bytes length |
| VarSize | String which contains Size of the UEFI Variable – 8 bytes length |
| VarDataBytes | String which contains variable data in bytes by bytes  ‘Byte0Byte2….ByteSize-1’ |

# UEFI Get / Set Variable’s Request / Response buffer: (can be located from c:\xmlcli\out directory)

**Note:** Currently output is dumped into binary file and below is the buffer format. XmlCli team is working to dump the output into command console itself and redirect to .txt file as well.

### Buffer Format

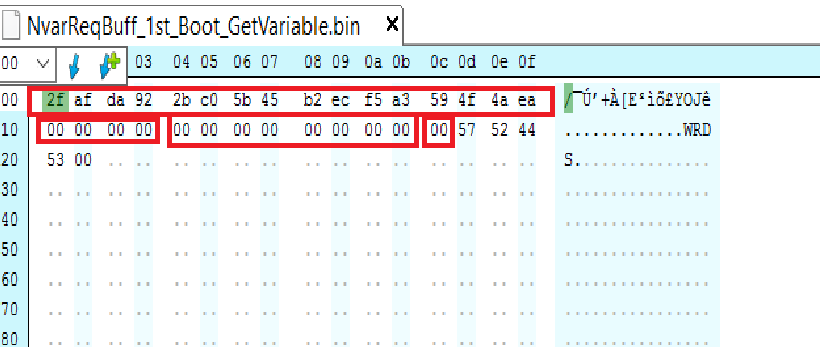


Here is some brief on each filed:

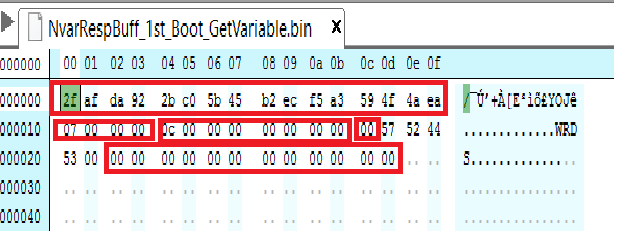
1. First 16 bytes are indicating GUID of UEFI Variable
2. Next 4 bytes are indicating attributes of UEFi variable
3. Next 8 bytes are indicating size of UEFI variable
4. Next 1 byte indicates the operation such as 00-Get,01-Set,02- GMS
5. Next null terminated string indicates the name of UEFI variable
6. Rest of bytes are indicating the data of UEFI variable

Note: only 1, 4, 5 are valid and rest 2, 3, 6 are ignorable in request buffer

### Request Buffer

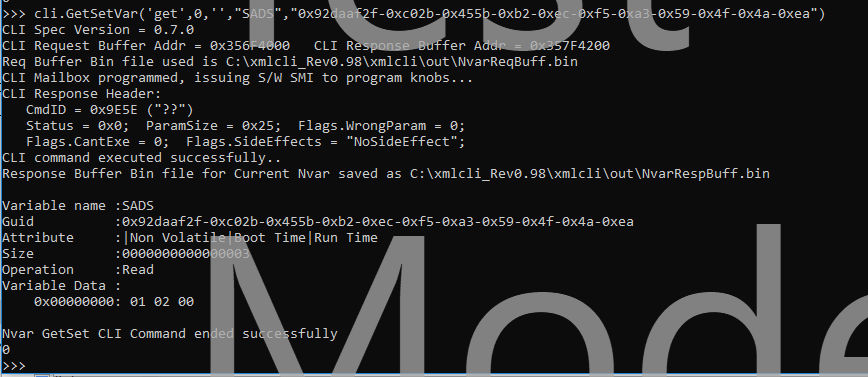


### Response Buffer

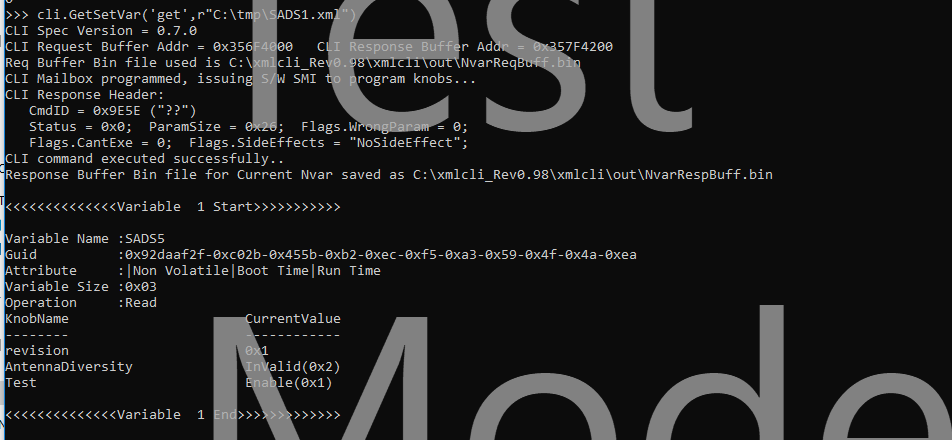


## Display the Response Buffer onto Console Screen

### Getting Variable Output through command line arguments:



### Getting variable output through Xml file as input arguments:



## Dump the Response Buffer onto .text file (Output.txt can be located from c:\xmlcli\out directory)

### Saving Variable Output through command line arguments:

Variable name :SADS

Guid :0x92daaf2f-0xc02b-0x455b-0xb2-0xec-0xf5-0xa3-0x59-0x4f-0x4a-0xea

Attribute :|Non Volatile|Boot Time|Run Time

Size :0x0000000000000003

Operation :Read

Variable Data :

0x00000000: 01 02 00

### Saving variable output through Xml file as input arguments:

<<<<<<<<<<<<<<Variable 1 Start>>>>>>>>>>>

Variable Name :SADS5

Guid :0x92daaf2f-0xc02b-0x455b-0xb2-0xec-0xf5-0xa3-0x59-0x4f-0x4a-0xea

Attribute :|Non Volatile|Boot Time|Run Time

Variable Size :0x03

Operation :Read

KnobName CurrentValue

-------- ------------

revision 0x1

AntennaDiversity InValid(0x2)

Test Enable(0x1)

<<<<<<<<<<<<<<Variable 1 End>>>>>>>>>>>>>

# Examples to access the UEFI variables

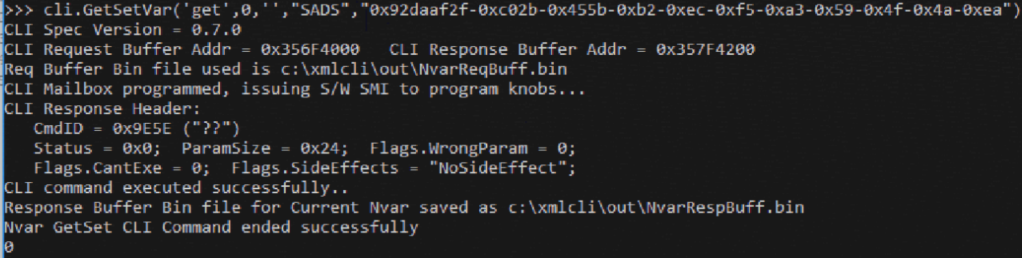
# CNV

## Get UEFI Variables

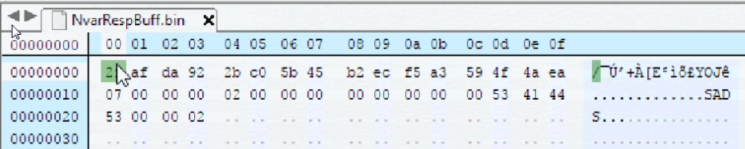
* Run the below command to get SADS variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

*cli.GetSetVar ( ‘get’,0, ‘’,”SADS”,” 0x92daaf2f-0xc02b-0x455b-0xb2-0xec-0xf5-0xa3-0x59-0x4f-0x4a-0xea”)*

*Note: Please make sure that command return value is zero, command has not been completed properly if you returns non-zero*



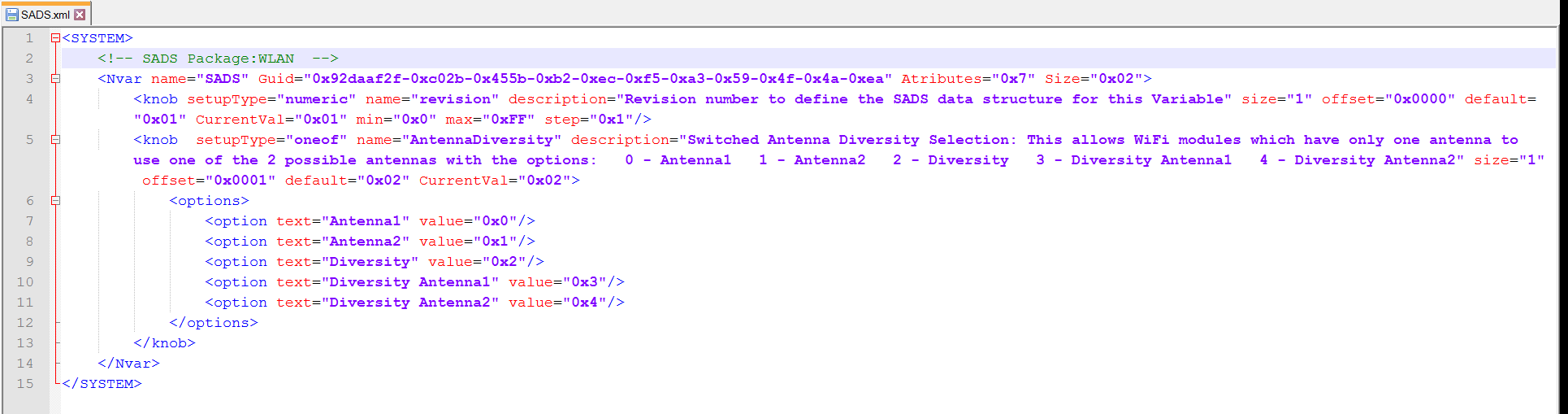
NvarRespBuff.bin:



* You can also use xml file to access the UEFI variable instead of giving values in command line as like below:

*cli.GetSetVar ( ‘get’,r”c:\xmlcli\SADS.xml”)*

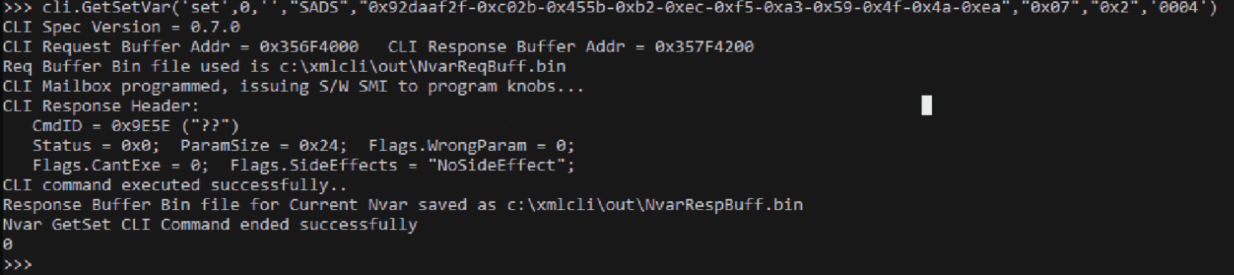
*SADS.xml:*



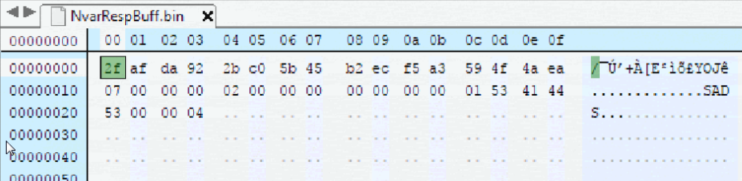
## Set UEFI Variables

* Run the below command to get SADS variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

*cli.GetSetVar ( ‘set’,0, ‘’,”SADS”,” 0x92daaf2f-0xc02b-0x455b-0xb2-0xec-0xf5-0xa3-0x59-0x4f-0x4a-0xea”,”0x07”,”0x02”,’0004’)*



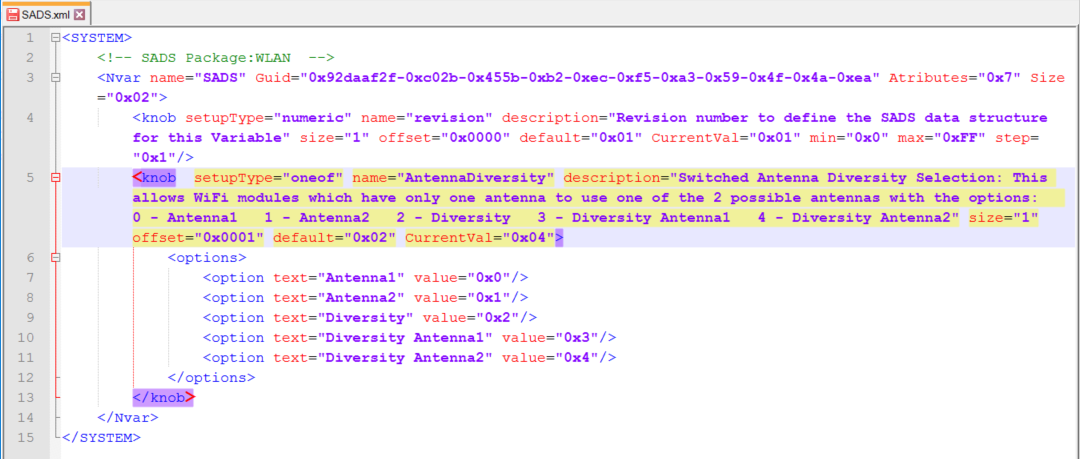
NvarRespBuff.bin:



* You can also use xml file to access the UEFI variable instead of giving values in command line as like below:

*cli.GetSetVar ( ‘set’,r”c:\xmlcli\SADS.xml”)*

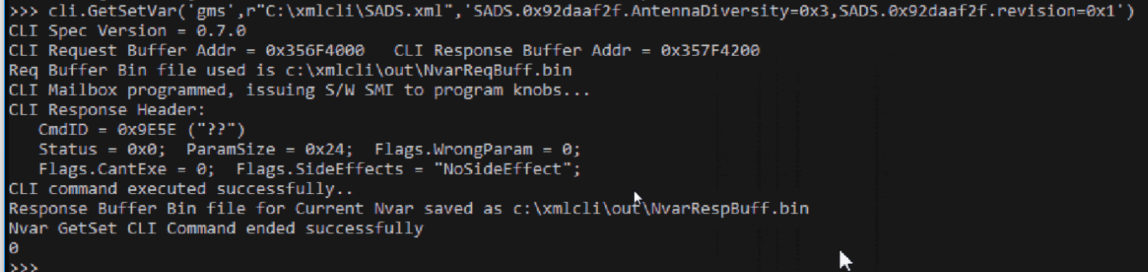
*SADS.xml:*



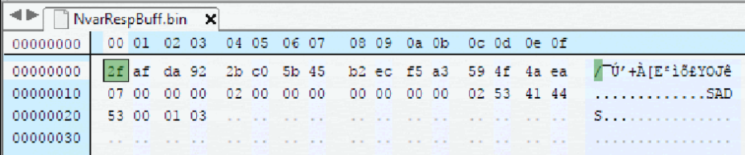
## Get and Modify the UEFI Variables

* Run the below command to get SADS variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

cli.GetSetVar ( ‘gms’,r” *c:\xmlcli\SADS*.xml”,’ SADS. 0x92daaf2f. AntennaDiversity =0x03, SADS. 0x92daaf2f. revision =0x1)



NvarRespBuff.bin:



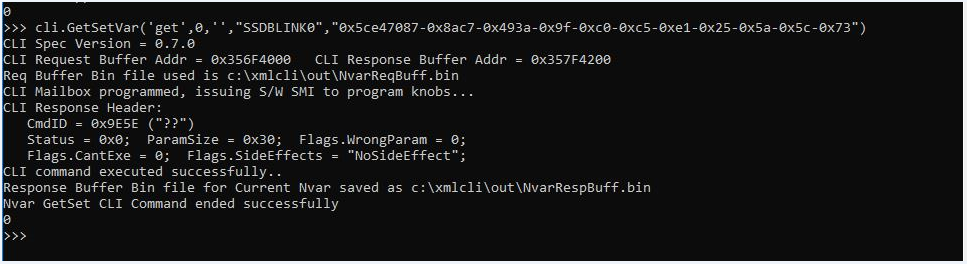
# MIPI Camera Configuration

## Get UEFI Variables

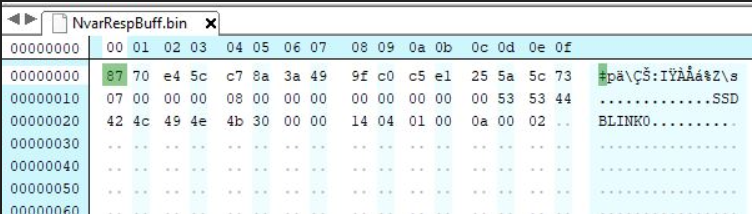
* Run the below command to get *SAPTD* variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

*cli.GetSetVar ( ‘get’,0, ‘’,”SSDBLINK0”,” 0x5ce47087-0x8ac7-0x493a-0x9f-0xc0-0xc5-0xe1-0x25-0x5a-0x5c-0x73”)*

*Note: Please make sure that command return value is zero, command has not been completed properly if you returns non-zero*

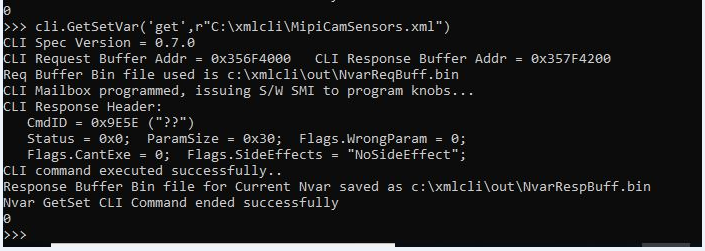


NvarRespBuff.bin:



* You can also use xml file to access the UEFI variable instead of giving values in command line as like below:

*cli.GetSetVar ( ‘get’,r”c:\xmlcli\ MipiCamSensors.xml”)*

**

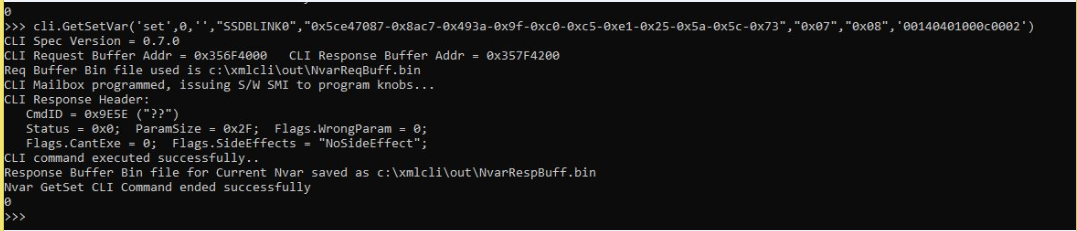
*MipiCamSensors.xml:*



## Set UEFI Variables

* Run the below command to get *SAPTD* variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

*cli.GetSetVar ( ‘Set’,0, ‘’, ”SSDBLINK0”,” 0x5ce47087-0x8ac7-0x493a-0x9f-0xc0-0xc5-0xe1-0x25-0x5a-0x5c-0x73”,”0x07”,”0x08”,’00140401000c0002’)*

**

* You can also use xml file to access the UEFI variable instead of giving values in command line as like below:

*cli.GetSetVar ( ‘set’,r”c:\xmlcli\ MipiCamSensors.xml”)*

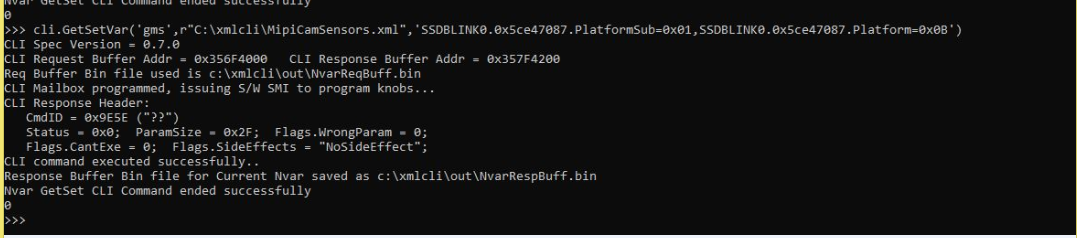
*MipiCamSensors.xml:*



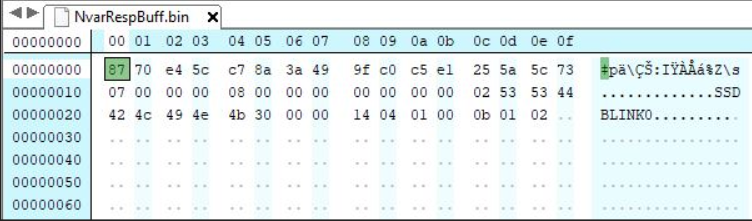
## Get and Modify the UEFI Variables

* Run the below command to get *SAPTD* and S2TDP variables and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

cli.GetSetVar ( ‘gms’,r” *c:\xmlcli\ MipiCamSensorsss*.xml”,’ *SSDBLINK0*. 0x5ce47087. PlatformSub=0x01, *SSDBLINK0*. 0x5ce47087. Platform=0x08)



NvarRespBuff.bin:



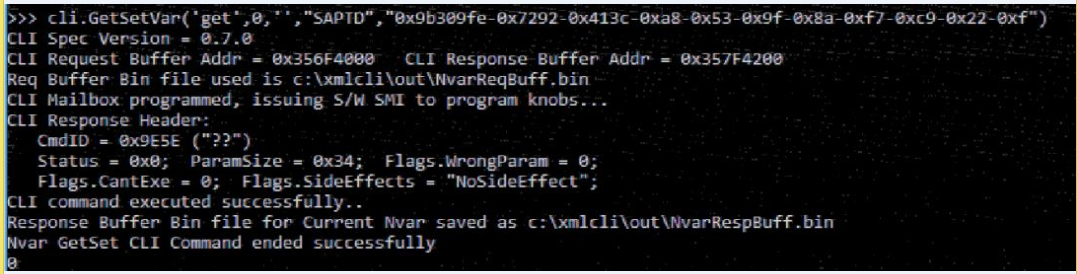
# DPTF

## Get UEFI Variables

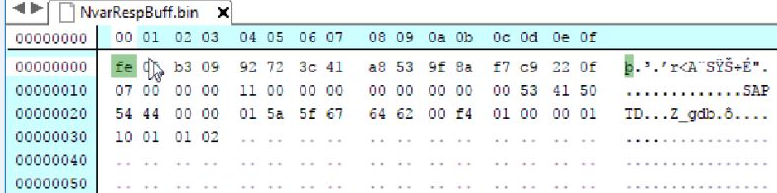
* Run the below command to get *SAPTD* variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

*cli.GetSetVar ( ‘get’,0, ‘’,”SAPTD”,” 0x9b309fe-0x7292-0x413c-0xa8-0x53-0x9f-0x8a-0xf7-0xc9-0x22-0xf”)*

*Note: Please make sure that command return value is zero, command has not been completed properly if you returns non-zero*



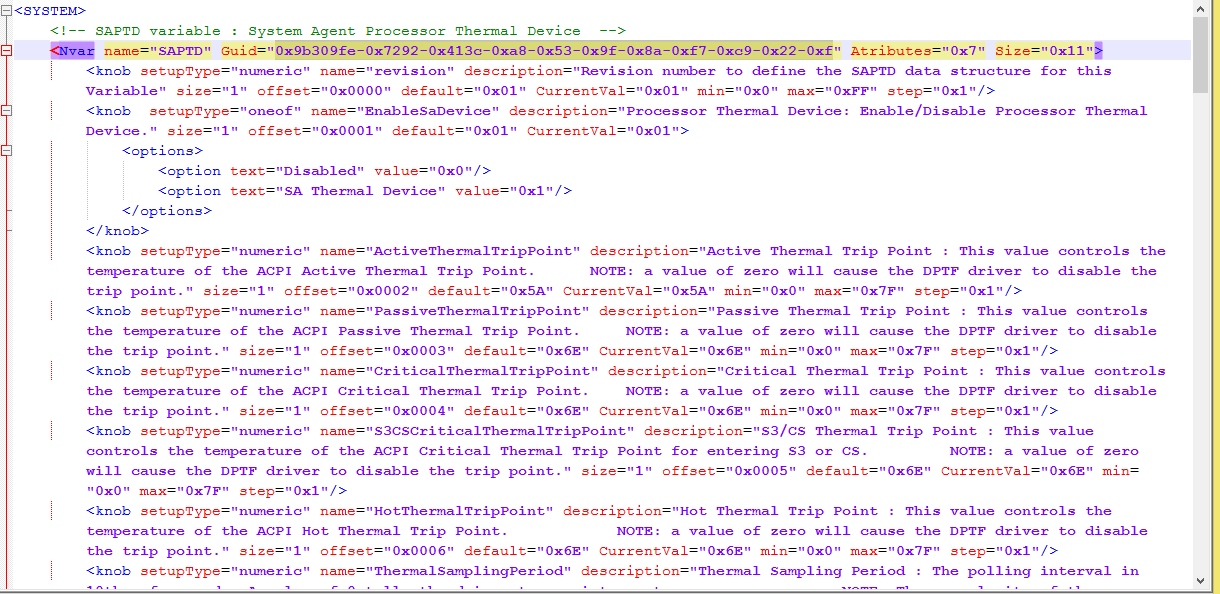
NvarRespBuff.bin:



* You can also use xml file to access the UEFI variable instead of giving values in command line as like below:

*cli.GetSetVar ( ‘get’,r”c:\xmlcli\ DptfUefiVariables.xml”)*

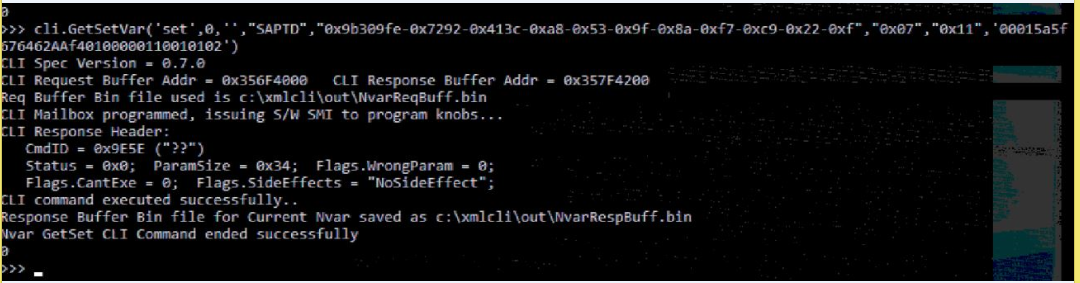
*DptfUefiVariables.xml:*



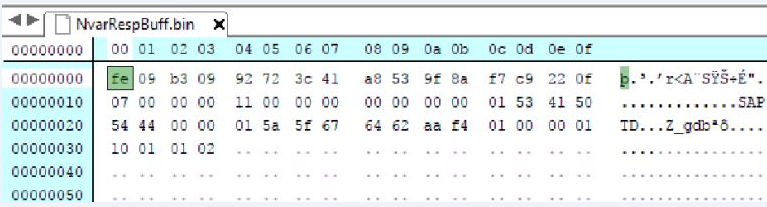
## Set UEFI Variables

* Run the below command to get *SAPTD* variable and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

*cli.GetSetVar ( ‘Set’,0, ‘’,”* *SAPTD”,” 0x9b309fe-0x7292-0x413c-0xa8-0x53-0x9f-0x8a-0xf7-0xc9-0x22-0xf”,”0x07”,”0x11”,’00015a5f676462AAf40100000110010102’)*



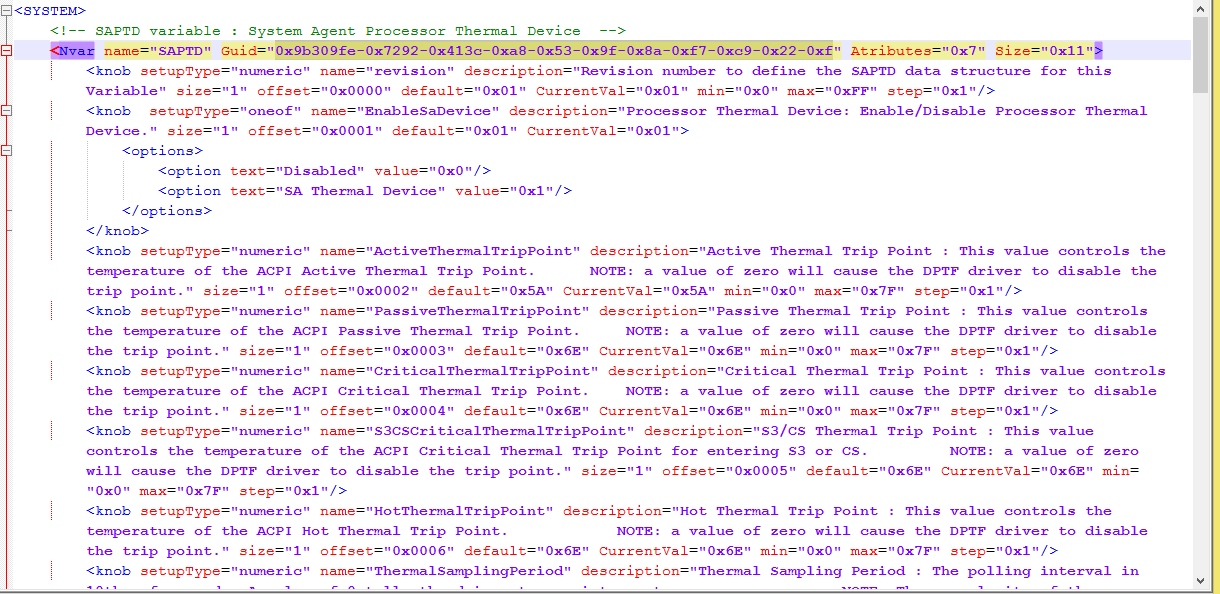
NvarRespBuff.bin:



* You can also use xml file to access the UEFI variable instead of giving values in command line as like below:

*cli.GetSetVar ( ‘set’,r”c:\xmlcli\DptfUefiVariables.xml”)*

*DptfUefiVariables.xml:*



## Get and Modify the UEFI Variables

* Run the below command to get *SAPTD* and S2TDP variables and return buffer will be available in *<Path\_To\_XmlCliRefScripts>/out/NvarRespBuff.bin*

cli.GetSetVar ( ‘gms’,r” *c:\xmlcli\ DptfUefiVariables*.xml”,’ *SAPTD*. 0x9b309fe. ThermalSamplingPeriod =0x03, S2TDP. 0x9b309fe. HotThermalTripPoint =0x1)