Security Guide

Keysight M9381A PXIe VSG, M9380A PXIe CW Source, and M9391A PXIe VSA

Notice: This document contains references to Agilent. Please note that Agilent's Test and Measurement business has become Keysight Technologies. For more information, go to www.keysight.com.

KEYSIGHT TECHNOLOGIES

Notices

Manual Part Number

M9300-90021

Edition

October2014, build 2014.10.25.1230

Purpose of this Document

This document details the internal memory locations of the instrument and describes instrument security features and the steps to declassify an instrument through memory sanitization or removal. For additional information on a particular product, the Keysight Instrument Security Database may be accessed here: www.keysight.com/find/security. For general information, the Keysight Aerospace and Defense web page may be found here: www.keysight.com/find/ad.

Sales and Technical Support

For product specific information and support, and to obtain the latest software and documentation, refer to the following Keysight web resources:

- www.keysight.com/find/M9381A (PXIe VSG)
- www.keysight.com/find/M9380A (PXIe CW Source)
- http://www.keysight.com/find/M9391A (PXIe VSA)
- www.keysight.com/find/M9300A (PXIe Freq. Reference)

Worldwide contact information for repair and service can be found at www.keysight.com/find/assist.

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Terms and Definitions

Clearing: Clearing is the process of eradicating the data on media before reusing the media so that the data can no longer be retrieved using the standard interfaces on the instrument. Clearing is typically used when the instrument is to remain in an environment with an acceptable level of protection. As defined in Section 8-301a of DOD 5220.22-M, “National Industrial Security Program Operating Manual (NISPOM)”,

Instrument Declassification: A term that refers to procedures that must be undertaken before an instrument can be removed from a secure environment, such as is the case when the instrument is returned for calibration. Declassification procedures include memory sanitization or memory removal, or both. Keysight declassification procedures are designed to meet the requirements specified in DOD 5220.22-M (NISPOM), Chapter 8.

Sanitization: Sanitization is the process of removing or eradicating stored data so that the data cannot be recovered using any known technology. Instrument sanitization is typically required when an instrument is moved from a secure to a non-secure environment such as when it is returned to the factory for calibration. (The instrument is declassified.) Keysight memory sanitization procedures are designed for customers who need to meet the requirements specified by the US Defense Security Service (DSS). These requirements are outlined in the “Clearing and Sanitization Matrix” issued by the Cognizant Security Agency (CSA) and referenced in National Industrial Security Program Operating Manual (NISPOM) DOD 5220.22-M ISL 01L-1 section 8-301.

Secure Erase: Secure Erase is a term that is used to refer to either the clearing or sanitization features of Keysight instruments.

References

DOD 5220.22-M, “National Industrial Security Program Operating Manual (NISPOM)”, United States Department of Defense. May be downloaded from: www.dss.mil/isp/fac_clear/download_nispom.html.

ODAA Process Guide for C&A of Classified Systems under NISPOM, Defense Security Service. DSS-cleared industries may request a copy of this document by following the instructions at: http://www.dss.mil/isp/odaa/request.html.

Contents

Product Memory Sanitization

Sanitization processes for the following Keysight product models are covered by this document:

- Multi-module instruments:
  - **M9381A** PXIe Vector Signal Generator drivers
  - **M9380A** PXIe CW Source instrument drivers
  - **M9391A** PXIe Vector Signal Generator drivers
- PXIe modules:
  - **M9300A** PXIe Frequency Reference
  - **M9301A** PXIe Synthesizer
  - **M9310A** PXIe Source Output
  - **M9311A** PXIe Digital Vector Modulator
  - **M9350A** PXIe RF Downconverter
  - **M9214A** PXIe IF Digitizer

M9381A PXIe Vector Signal Generator and M9380A PXIe CW Source Instrument Drivers

Both of these products use the AgM938x driver. The driver installs the IVI-C, IVI-COM, LabView, and MATLAB driver components, as well as the soft front panel and kernel device driver on your controller.

Memory Type: Controller Hard Drive	Memory Size: unknown
Memory Function: Stores device drivers, example programs, example waveforms, help system, and user documentation.	
User Modifiable? Yes	Volatile? No
Memory Erase Processes: To uninstall the AgM938x instrument driver from the controller, perform the relevant procedure below. Windows 7: 1. Select Start > Control Panel > Programs and Features 2. Select Keysight M938x 3. Select Uninstall Windows XP: 1. Select Start > Control Panel 2. Double-click Add or Remove Programs 3. Select Keysight M938x 4. Select Remove To clear all information from the controller used with the M9381A PXIe Vector Signal Generator and M9380A PXIe CW Source, follow the memory erase procedure for the controller as recommended by the manufacturer.	

M9391A PXIe Vector Signal Analyzer Drivers

This product uses the same AgM9391 driver. The driver installs the IVI-C, IVI-COM, LabView, and MATLAB driver components, as well as the soft front panel and kernel device driver on your controller.

Memory Type: Controller Hard Drive	**Memory Size**: unknown
Memory Function: Stores device drivers, example programs, example waveforms, help system, and user documentation.	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To uninstall the AgM9391 instrument driver from the controller, perform the relevant procedure below. Windows 7: 1. Select **Start > Control Panel > Programs and Features** 2. Select **Keysight M9391** 3. Select **Uninstall** Windows XP: 1. Select **Start > Control Panel** 2. Double-click **Add or Remove Programs** 3. Select **Keysight M9391** 4. Select **Remove** To clear all information from the controller used with the M9391A PXIe Vector Signal Analyzer, follow the memory erase procedure for the controller as recommended by the manufacturer.	

M9300A PXIe Frequency Reference

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores FPGA firmware, carrier board information, security settings and backup of security settings.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: Can be overwritten using the Keysight Soft Front Panel firmware update utility; cannot be easily erased or corrupted by the user.	

Memory Type: Flash Memory	**Memory Size**: 64 M Bit
Memory Function: Stores FPGA firmware, slug board information, security settings and backup of security settings.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: Can be overwritten using the Keysight Soft Front Panel firmware update utility; cannot be easily erased or corrupted by the user.	

M9301A PXIe Synthesizer

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores Module Model Number, Serial Number, Manufacturing Number, Options, PCB Part and Version Numbers, Cal Verify Date, Max Module Temperature, and Calibration and Alignment Data.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Device firmware. Images can be changed using the Keysight soft front panel firmware update utility.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores calibration preferences: due date subject to periodic cal, module cal warnings, cal due reminder, module cal reminder and passphrase	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: You can clear the passphrase by using the relevant IVI driver code: • If the module is used in an M9381A instrument, see "M9381A Memory Clear Code" (page 14). • If the module is used in an M9380A instrument, see "M9380A Memory Clear Code" (page 16). • If the module is used in an M9391A instrument, see "M9391A Memory Clear Code" (page 18).	

Memory Type: FPGA	**Memory Size**:
Memory Function: Frequency start/stop/step, power, waveform, and impairments.	
User Modifiable? Yes	**Volatile?** Yes
Memory Erase Processes: Cycle power	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores user customizable asset number and system identification	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: You can clear the asset number and system identification values by using the relevant IVI driver code: • If the module is used in an M9381A instrument, see "M9381A Memory Clear Code" (page 14). • If the module is used in an M9380A instrument, see "M9380A Memory Clear Code" (page 16). • If the module is used in an M9391A instrument, see "M9391A Memory Clear Code" (page 18).	

M9310A PXIe Source Output

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores module model number, serial number, manufacturing number, options, PCB part and version numbers, cal verify date, max module temperature, and calibration and alignment data.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Device firmware. Images can be changed using the Keysight Soft Front Panel firmware update utility.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores calibration preferences: due date subject to periodic cal, module cal warnings, cal due reminder, module cal reminder and passphrase	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: You can clear the passphrase by using the relevant IVI driver code: • If the module is used in an M9381A instrument, see "M9381A Memory Clear Code" (page 14). • If the module is used in an M9380A instrument, see "M9380A Memory Clear Code" (page 16).	

Memory Type: FPGA	**Memory Size**:
Memory Function: Frequency start/stop/step, power, waveform, and impairments.	
User Modifiable? Yes	**Volatile?** Yes
Memory Erase Processes: Cycle power	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores user customizable asset number and system identification	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: You can clear the asset number and system identification values by using the relevant IVI driver code: • If the module is used in an M9381A instrument, see "M9381A Memory Clear Code" (page 14). • If the module is used in an M9380A instrument, see "M9380A Memory Clear Code" (page 16).	

M9311A PXIe Digital Vector Modulator

(NOTE: The M9311A PXIe Digital Vector Modulator is only used with M9381A PXIe Vector Signal Generator.)

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores module model number, serial number, manufacturing number, options, PCB part and version numbers, cal verify date, max module temperature, and calibration and alignment data.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Device firmware. Images can be changed using the Keysight Soft Front Panel firmware update utility.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores calibration preferences: due date subject to periodic cal, module cal warnings, cal due reminder, module cal reminder and passphrase	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To clear the passphrase, use the IVI driver code provided in "M9381A Memory Clear Code" (page 14).	

Memory Type: FPGA	**Memory Size**:
Memory Function: Frequency Start/Stop/Step, Power, Waveform, and Impairments.	
User Modifiable? Yes	**Volatile?** Yes
Memory Erase Processes: Cycle power	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores User Customizable Asset Number and System Identification	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To clear the asset number and system identification values, use the IVI driver code provided in "M9381A Memory Clear Code" (page 14).	

M9350A PXIe RF Downconverter

(NOTE: The M9350A PXIe Downconverter is only used with M9391A PXIe Vector Signal Analyzer.)

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores module model number, serial number, manufacturing number, options, PCB part and version numbers, cal verify date, max module temperature, and calibration and alignment data.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Device firmware. Images can be changed using the Keysight Soft Front Panel firmware update utility.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores calibration preferences: due date subject to periodic cal, module cal warnings, cal due reminder, module cal reminder and passphrase	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To clear the passphrase, use the IVI driver code provided in "M9391A Memory Clear Code" (page 18).	

Memory Type: FPGA	**Memory Size**:
Memory Function: Stores: IQ, spectrum and power settings; advanced options; dither; alignments.	
User Modifiable? Yes	**Volatile?** Yes
Memory Erase Processes: Cycle power	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores User Customizable Asset Number and System Identification	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To clear the asset number and system identification values, use the IVI driver code provided in "M9391A Memory Clear Code" (page 18).	

M9214A PXIe IF Digitizer

(NOTE: The M9214A PXIe IF Digitizer is only used with M9391A PXIe Vector Signal Analyzer.)

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores module model number, serial number, manufacturing number, options, PCB part and version numbers, cal verify date, max module temperature, and calibration and alignment data.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Device firmware. Images can be changed using the Keysight Soft Front Panel firmware update utility.	
User Modifiable? No	**Volatile?** No
Memory Erase Processes: None, this is not user accessible	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores calibration preferences: due date subject to periodic cal, module cal warnings, cal due reminder, module cal reminder and passphrase	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To clear the passphrase, use the IVI driver code provided in "M9391A Memory Clear Code" (page 18).	

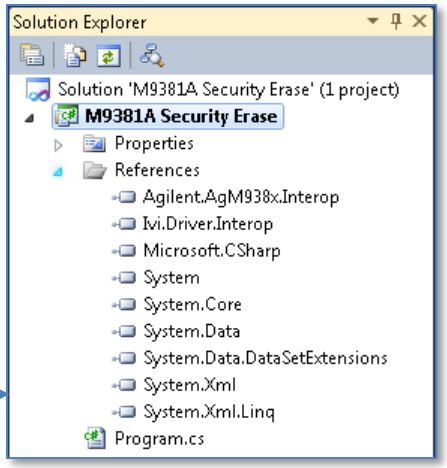
Memory Type: FPGA	**Memory Size**:
Memory Function: Stores: IQ, spectrum and power settings; advanced options; dither; alignments.	
User Modifiable? Yes	**Volatile?** Yes
Memory Erase Processes: Cycle power	

Memory Type: Flash Memory	**Memory Size**: 128 M Bit
Memory Function: Stores User Customizable Asset Number and System Identification	
User Modifiable? Yes	**Volatile?** No
Memory Erase Processes: To clear the asset number and system identification values, use the IVI driver code provided in "M9391A Memory Clear Code" (page 18).	

M9381A Memory Clear Code

Below is the IVI code to clear the memory from the M9381A PXIe Vector Signal Generator and its modular components (M9300A Reference, M9301A Synthesizer, M9310A Source Output, and M9311A Modulator). The procedures in this code sample clear the Asset Number, System ID, and Cal passphrase from the flash memory.

All you need to do is copy and paste the code into a console application and include the correct driver references – see inset picture (right)



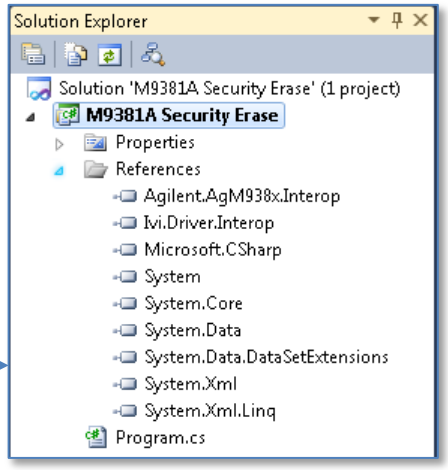

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Ivi.Driver.Interop;
using Agilent.AgM938x.Interop;

namespace M9381A_Security_Erase
{
    class Program
    {
        static void Main(string[] args)
        {

            //running this program will clear the flash memory of the M9381A Vector Signal Generator
            //The flash memory cleared is the Asset Number, System ID, and the passphrase protecting the calibration preferences
            //ONLY run this program if you are sure you want to clear this information

            //initialize the driver
            IAgM938x m9381a = new AgM938x();
            string resource = ""; //enter in the VISA resource between the quotes for the instrument getting cleared
            string options = "QueryInstrStatus=true, Simulate=false, DriverSetup=Trace=false";
            bool idquery = true;
            bool reset = true;
            m9381a.Initialize(resource, idquery, reset, options);
            Console.WriteLine("Driver Initialized.\n Press enter to continue\n");
            Console.ReadLine();

            //test to write to modules. It is commented out because it does not need to be run to clear the memory
            //m9381aWrite(m9381a.Modules.get_Item("M9300A"));
            //m9381aWrite(m9381a.Modules.get_Item("M9301A"));
            //m9381aWrite(m9381a.Modules.get_Item("M9310A"));
            //m9381aWrite(m9381a.Modules.get_Item("M9311A"));

            //Read back asset numbers and system ID from each module
            string refAsset = m9381a.Modules.get_Item("M9300A").Nonvolatile.AssetNumber;
            string refID = m9381a.Modules.get_Item("M9300A").Nonvolatile.SystemIdentification;
            string synthAsset = m9381a.Modules.get_Item("M9301A").Nonvolatile.AssetNumber;
            string synthID = m9381a.Modules.get_Item("M9301A").Nonvolatile.SystemIdentification;
            string outputAsset = m9381a.Modules.get_Item("M9310A").Nonvolatile.AssetNumber;
            string outputID = m9381a.Modules.get_Item("M9310A").Nonvolatile.SystemIdentification;
            string dvmAsset = m9381a.Modules.get_Item("M9311A").Nonvolatile.AssetNumber;
            string dvmID = m9381a.Modules.get_Item("M9311A").Nonvolatile.SystemIdentification;
            Console.WriteLine("Reference Asset is:" + refAsset + "\n");
            Console.WriteLine("Reference System ID is:" + refID + "\n");
            Console.WriteLine("Synthesizer Asset is:" + synthAsset + "\n");
            Console.WriteLine("Synthesizer System ID is:" + synthID + "\n");
            Console.WriteLine("Source Output Asset is:" + outputAsset + "\n");
            Console.WriteLine("Source Output ID is:" + outputID + "\n");
            Console.WriteLine("DVM Asset is:" + dvmAsset + "\n");
            Console.WriteLine("DVM System ID is:" + dvmID + "\n\n");

            //begin clear
            Console.WriteLine("Press Enter to Clear asset number and system ID");
            Console.ReadLine();

            //clear asset number and system ID and Calibration Preferences passphrase
            m9381aClear(m9381a.Modules.get_Item("M9300A"));
            m9381aClear(m9381a.Modules.get_Item("M9301A"));
            m9381aClear(m9381a.Modules.get_Item("M9310A"));
```

```
            m9381aClear(m9381a.Modules.get_Item("M9311A"));

            //read back module asset numbers and ID to verify memory clear
            Console.WriteLine("press enter to verify clear");
            Console.ReadLine();
            refAsset = m9381a.Modules.get_Item("M9300A").Nonvolatile.AssetNumber;
            refID = m9381a.Modules.get_Item("M9300A").Nonvolatile.SystemIdentification;
            synthAsset = m9381a.Modules.get_Item("M9301A").Nonvolatile.AssetNumber;
            synthID = m9381a.Modules.get_Item("M9301A").Nonvolatile.SystemIdentification;
            outputAsset = m9381a.Modules.get_Item("M9310A").Nonvolatile.AssetNumber;
            outputID = m9381a.Modules.get_Item("M9310A").Nonvolatile.SystemIdentification;
            dvmAsset = m9381a.Modules.get_Item("M9311A").Nonvolatile.AssetNumber;
            dvmID = m9381a.Modules.get_Item("M9311A").Nonvolatile.SystemIdentification;
            Console.WriteLine("Reference Asset No is:" + refAsset + "\n");
            Console.WriteLine("Reference System ID is:" + refID + "\n");
            Console.WriteLine("Synthesizer Asset No is:" + synthAsset + "\n");
            Console.WriteLine("Synthesizer System ID is:" + synthID + "\n");
            Console.WriteLine("Source Output Asset No is:" + outputAsset + "\n");
            Console.WriteLine("Source Output System ID is:" + outputID + "\n");
            Console.WriteLine("DVM Asset is:" + dvmAsset + "\n");
            Console.WriteLine("DVM System ID is:" + dvmID + "\n\n");
            Console.WriteLine("\n Memory clear complete, press enter to exit program");
            Console.ReadLine();

            //close the driver session
            m9381a.Close();
        }

        //test method to write to the modules. It is commented out because it does not need to be run to clear the memory
        //static void m9381aWrite(IAgM938xModule module)
        //{
        // module.Nonvolatile.Clear();
        // module.Nonvolatile.SystemIdentification = "system ID";
        // module.Nonvolatile.AssetNumber = "123456789";
        // string oldPassphrase = module.Nonvolatile.Passphrase;
        // module.Nonvolatile.Write(oldPassphrase);
        //}

        //method to clear the Passphrase and Asset Number/System ID of each module
        static void m9381aClear(IAgM938xModule module)
        {
            module.Nonvolatile.Clear();
            module.Nonvolatile.SystemIdentification = "";
            module.Nonvolatile.AssetNumber = "";
            string newPassphrase = "";
            string oldPassphrase = module.Nonvolatile.Passphrase;
            module.Nonvolatile.Passphrase = newPassphrase;
            module.Nonvolatile.Write(oldPassphrase);
        }
    }
}
```

M9380A Memory Clear Code

Below is the IVI code to clear the memory from the M9380A PXIe CW Source and its modular components (M9300A Reference, M9301A Synthesizer, and M9310A Source Output). The procedures in this code sample clear the Asset Number, System ID, and Cal passphrase from the flash memory.

All you need to do is copy and paste the code into a console application and include the correct driver references – see inset picture (right).



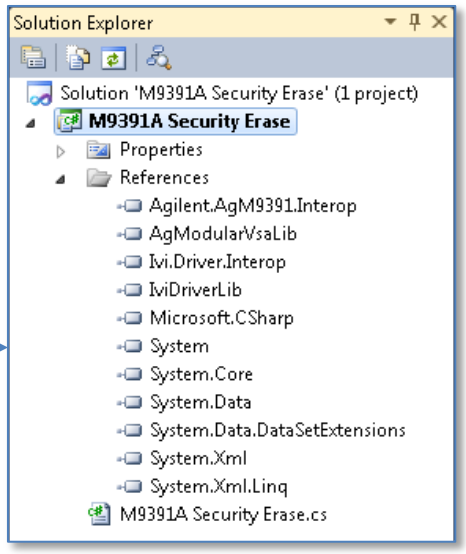

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Ivi.Driver.Interop;
using Agilent.AgM938x.Interop;

namespace M9380A_Security_Erase
{
    class Program
    {
        static void Main(string[] args)
        {

            //running this program will clear the flash memory of the M9380A CW Source
            //The flash memory cleared is the Asset Number, System ID, and the passphrase protecting the calibration preferences
            //ONLY run this program if you are sure you want to clear this information

            //initialize the driver
            IAgM938x m9380a = new AgM938x();
            string resource = ""; //enter in the VISA resource between the quotes for the instrument getting cleared
            string options = "QueryInstrStatus=true, Simulate=false, DriverSetup=Trace=false";
            bool idquery = true;
            bool reset = true;
            m9380a.Initialize(resource, idquery, reset, options);
            Console.WriteLine("Driver Initialized.\n Press enter to continue\n");
            Console.ReadLine();


            //test to write to modules. It is commented out because it does not need to be run to clear the memory
            //m9380aWrite(m9380a.Modules.get_Item("M9300A"));
            //m9380aWrite(m9380a.Modules.get_Item("M9301A"));
            //m9380aWrite(m9380a.Modules.get_Item("M9310A"));

            //Read back asset numbers and system ID from each module
            string refAsset = m9380a.Modules.get_Item("M9300A").Nonvolatile.AssetNumber;
            string refID = m9380a.Modules.get_Item("M9300A").Nonvolatile.SystemIdentification;
            string synthAsset = m9380a.Modules.get_Item("M9301A").Nonvolatile.AssetNumber;
            string synthID = m9380a.Modules.get_Item("M9301A").Nonvolatile.SystemIdentification;
            string outputAsset = m9380a.Modules.get_Item("M9310A").Nonvolatile.AssetNumber;
            string outputID = m9380a.Modules.get_Item("M9310A").Nonvolatile.SystemIdentification;
            Console.WriteLine("Reference Asset is:" + refAsset + "\n");
            Console.WriteLine("Reference System ID is:" + refID + "\n");
            Console.WriteLine("Synthesizer Asset is:" + synthAsset + "\n");
            Console.WriteLine("Synthesizer System ID is:" + synthID + "\n");
            Console.WriteLine("Source Output Asset is:" + outputAsset + "\n");
            Console.WriteLine("Source Output ID is:" + outputID + "\n");

            //begin clear
            Console.WriteLine("Press Enter to Clear asset number and system ID");
            Console.ReadLine();

            //clear asset number and system ID and Calibration Preferences passphrase
            m9380aClear(m9380a.Modules.get_Item("M9300A")); m9380aClear(m9380a.Modules.get_Item("M9301A"));
            m9380aClear(m9380a.Modules.get_Item("M9310A"));

            //read back module asset numbers and ID to verify memory clear
            Console.WriteLine("press enter to verify clear");
            Console.ReadLine();
            refAsset = m9380a.Modules.get_Item("M9300A").Nonvolatile.AssetNumber;
            refID = m9380a.Modules.get_Item("M9300A").Nonvolatile.SystemIdentification;
            synthAsset = m9380a.Modules.get_Item("M9301A").Nonvolatile.AssetNumber;
```

```
            synthID = m9380a.Modules.get_Item("M9301A").Nonvolatile.SystemIdentification;
            outputAsset = m9380a.Modules.get_Item("M9310A").Nonvolatile.AssetNumber;
            outputID = m9380a.Modules.get_Item("M9310A").Nonvolatile.SystemIdentification;
            Console.WriteLine("Reference Asset No is:" + refAsset + "\n");
            Console.WriteLine("Reference System ID is:" + refID + "\n");
            Console.WriteLine("Synthesizer Asset No is:" + synthAsset + "\n");
            Console.WriteLine("Synthesizer System ID is:" + synthID + "\n");
            Console.WriteLine("Source Output Asset No is:" + outputAsset + "\n");
            Console.WriteLine("Source Output System ID is:" + outputID + "\n");
            Console.WriteLine("\n Memory clear complete, press enter to exit program");
            Console.ReadLine();

            //close the driver session
            m9380a.Close();
        }


        //test method to write to the modules. It is commented out because it does not need to be run to clear the memory.
        //static void m9380aWrite(IAgM938xModule module)
        //{
        // module.Nonvolatile.Clear();
        // module.Nonvolatile.SystemIdentification = "system ID";
        // module.Nonvolatile.AssetNumber = "123456789";
        // string oldPassphrase = module.Nonvolatile.Passphrase;
        // module.Nonvolatile.Write(oldPassphrase);
        //}

        //method to clear the Passphrase and Asset Number/System ID of each module
        static void m9380aClear(IAgM938xModule module)
        {
            module.Nonvolatile.Clear();
            module.Nonvolatile.SystemIdentification = "";
            module.Nonvolatile.AssetNumber = "";
            string newPassphrase = "";
            string oldPassphrase = module.Nonvolatile.Passphrase;
            module.Nonvolatile.Passphrase = newPassphrase;
            module.Nonvolatile.Write(oldPassphrase);
        }
    }
}
```

M9391A Memory Clear Code

Below is the IVI code to clear the memory from the M9391A PXIe Vector Signal Analyzer and its modular components (M9300A Reference, M9301A Synthesizer, M9350A Downconverter, and M9214A Digitizer). The procedures in this code sample clear the Asset Number, System ID, and Cal passphrase from the flash memory.

All you need to do is copy and paste the code into a console application and include the correct driver references – see inset picture (right).




```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Ivi.Driver.Interop;
using Agilent.AgM9391.Interop;

namespace M9391A_Security_Erase
{
    class Program
    {
        static void Main(string[] args)
        {

            //Running this program will clear the flash memory of the M9391A Vector Signal Analyzer multi-module instrument.
            //The flash memory cleared is the Asset Number, System ID, and the passphrase protecting the calibration preferences.
            //ONLY run this program if you are sure you want to clear this information.

            //initialize the driver
            IAgM9391 m9391a = new AgM9391();
            string resource = "";   //enter in the VISA resource between the quotes for the instrument getting cleared
            string options = "QueryInstrStatus=true, Simulate=false, DriverSetup=Trace=false";
            bool idquery = true;
            bool reset = true;
            m9391a.Initialize(resource, idquery, reset, options);
            Console.WriteLine("Driver Initialized.\n Press enter to continue\n");
            Console.ReadLine();

            //Test to write to modules. It is commented out because it does not need to be run to clear the memory.
            //m9391aWrite(m9391a.Modules.get_Item("M9300A"));
            //m9391aWrite(m9391a.Modules.get_Item("M9301A"));
            //m9391aWrite(m9391a.Modules.get_Item("M9350A"));
            //m9391aWrite(m9391a.Modules.get_Item("M9214A"));

            //Read back asset numbers and system ID from each module
            string refAsset = m9391a.Modules.get_Item("M9300A").Nonvolatile.AssetNumber;
            string refID = m9391a.Modules.get_Item("M9300A").Nonvolatile.SystemIdentification;
            string synthAsset = m9391a.Modules.get_Item("M9301A").Nonvolatile.AssetNumber;
            string synthID = m9391a.Modules.get_Item("M9301A").Nonvolatile.SystemIdentification;
            string DCAsset = m9391a.Modules.get_Item("M9350A").Nonvolatile.AssetNumber;
            string DCID = m9391a.Modules.get_Item("M9350A").Nonvolatile.SystemIdentification;
            string digAsset = m9391a.Modules.get_Item("M9214A").Nonvolatile.AssetNumber;
            string digID = m9391a.Modules.get_Item("M9214A").Nonvolatile.SystemIdentification;
            Console.WriteLine("Reference Asset is:" + refAsset + "\n");
            Console.WriteLine("Reference System ID is:" + refID + "\n");
            Console.WriteLine("Synthesizer Asset is:" + synthAsset + "\n");
            Console.WriteLine("Synthesizer System ID is:" + synthID + "\n");
            Console.WriteLine("DownConverter Asset is:" + DCAsset + "\n");
            Console.WriteLine("DownConverter System ID is:" + DCID + "\n");
            Console.WriteLine("Digitizer Asset is:" + digAsset + "\n");
            Console.WriteLine("Digitzer System ID is:" + digID + "\n\n");

            //Begin clear
            Console.WriteLine("Press Enter to Clear asset number and system ID");
            Console.ReadLine();

            //Clear asset number and system ID and Calibration Preferences passphrase.
            m9391aClear(m9391a.Modules.get_Item("M9300A"));
            m9391aClear(m9391a.Modules.get_Item("M9301A"));
            m9391aClear(m9391a.Modules.get_Item("M9350A"));
            m9391aClear(m9391a.Modules.get_Item("M9214A"));
```

```
            //Read back module asset numbers and ID to verify memory clear.
            Console.WriteLine("press enter to verify clear");
            Console.ReadLine();
            refAsset = m9391a.Modules.get_Item("M9300A").Nonvolatile.AssetNumber;
            refID = m9391a.Modules.get_Item("M9300A").Nonvolatile.SystemIdentification;
            synthAsset = m9391a.Modules.get_Item("M9301A").Nonvolatile.AssetNumber;
            synthID = m9391a.Modules.get_Item("M9301A").Nonvolatile.SystemIdentification;
            DCAsset = m9391a.Modules.get_Item("M9350A").Nonvolatile.AssetNumber;
            DCID = m9391a.Modules.get_Item("M9350A").Nonvolatile.SystemIdentification;
            digAsset = m9391a.Modules.get_Item("M9214A").Nonvolatile.AssetNumber;
            digID = m9391a.Modules.get_Item("M9214A").Nonvolatile.SystemIdentification;
            Console.WriteLine("Reference Asset No is:" + refAsset + "\n");
            Console.WriteLine("Reference System ID is:" + refID + "\n");
            Console.WriteLine("Synthesizer Asset No is:" + synthAsset + "\n");
            Console.WriteLine("Synthesizer System ID is:" + synthID + "\n");
            Console.WriteLine("DownConverter Asset No is:" + DCAsset + "\n");
            Console.WriteLine("DownConverter System ID is:" + DCID + "\n");
            Console.WriteLine("Digitizer Asset is:" + digAsset + "\n");
            Console.WriteLine("Digitzer System ID is:" + digID + "\n\n");
            Console.WriteLine("\n Memory clear complete, press enter to exit program");
            Console.ReadLine();

            //Close the driver session.
            m9391a.Close();
        }

        //Test method to write to the modules. It is commented out because it does not need to be run to clear the memory.
        //static void m9391aWrite(IAgM9391Module module)
        //{
        //   module.Nonvolatile.Clear();
        //   module.Nonvolatile.SystemIdentification = "system ID";
        //   module.Nonvolatile.AssetNumber = "123456789";
        //   string oldPassphrase = module.Nonvolatile.Passphrase;
        //   module.Nonvolatile.Write(oldPassphrase);
        //}

        //Method to clear the Passphrase and Asset Number/System ID of each module.
        static void m9391aClear(IAgM9391Module module)
        {
                  module.Nonvolatile.Clear();
                  module.Nonvolatile.SystemIdentification = "";
                  module.Nonvolatile.AssetNumber = "";
                  string newPassphrase = "";
                  string oldPassphrase = module.Nonvolatile.Passphrase;
                  module.Nonvolatile.Passphrase = newPassphrase;
                  module.Nonvolatile.Write(oldPassphrase);
        }
    }
}
```


KEYSIGHT
TECHNOLOGIES