

Startup Guide

Keysight M90XA X-Series Measurement Applications for Modular Instruments

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.

Notices

© Keysight Technologies, Inc. 2013–2015

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies, Inc. as governed by United States and international copyright laws.

Manual Part Number

M9000-90001

Edition

April 2015, build 2015.04.01.1230

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.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement (“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at <http://www.keysight.com/find/sweula>. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A **WARNING** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a **WARNING** notice until the indicated conditions are fully understood and met.

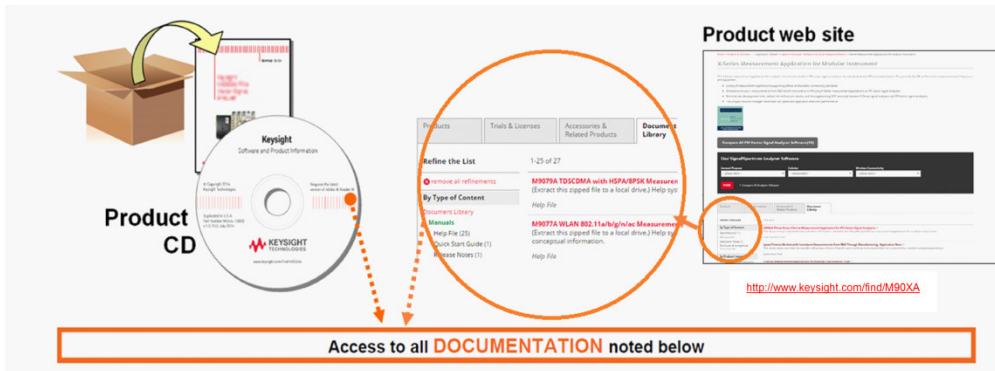
Contents

Chapter 1 – Documentation Map	6
Chapter 2 – Introduction	7
2.1 Overview of M90XA Software	7
2.1.1 X-Series Measurement Applications for Modular Instruments.....	7
2.2 Related Documentation	8
Chapter 3 – Installation and License Setup	11
3.1 System Requirements.....	11
3.1.1 Setup Notes	11
3.2 Installing M90XA Software	12
3.2.1 Prerequisites.....	12
3.2.2 M90XA Software Installer.....	12
3.2.3 Installation Procedure	12
3.3 Setting up Licenses.....	14
3.3.1 M90XA Software Licensing	14
Chapter 4 – Launching M90XA.....	17
4.1 Startup Options.....	20
Chapter 5 – Using M90XA Software	23
5.1 High Level Architecture	24
5.1.1 M90XA Application Container.....	24
5.1.2 M9000 Resource Manager.....	25
5.1.3 Supported VSA Hardware.....	25
5.1.4 M9391A and M9393A IVI Drivers.....	25
5.2 Using the GUI.....	25
5.3 Using SCPI.....	26
5.3.1 SCPI Control of M90XA	26

5.3.2	HiSLIP	28
5.3.3	SICL Server (VXI-11).....	28
5.4	Using a Modular VSA Versus a Swept Analyzer	29
5.4.1	Power Range.....	29
5.4.2	Spectrum Measurements	29
5.5	Triggering and Gating	31
5.5.1	Trigger Differences	31
5.6	Alignments.....	33
5.6.1	Turning Off Auto Alignments.....	33
5.6.2	M9393A Alignments	34
5.7	Event Logging	34
5.8	Optimizing Performance	35
5.8.1	PXIe Chassis Configuration	35
5.8.2	Controller Configuration.....	35
5.9	Additional Information	35
5.9.1	M9063A Analog Demod	35
Chapter 6 – Programming with M90XA.....		36
6.1	M90XA Example Programs	36
6.2	Shared Native .dlls.....	37
6.3	LabVIEW Example.....	38
6.3.1	Prerequisites.....	38
6.3.2	Executing the LabVIEW Example.....	38
6.3.3	The LabVIEW Example Code.....	39
6.4	MATLAB Example.....	40
6.4.1	Running the MATLAB Example	40
6.4.2	Code Walkthrough	41
6.5	C or C++ Example.....	44
6.5.1	Building the C/C++ Example.....	45
6.5.2	Running the C/C++ Example.....	45
6.5.3	Deploying a C/C++ Program	46
6.6	C (Sharp) Example	47

6.6.1	Code Walkthrough	49
6.7	C (Sharp) Example Form	52
6.7.1	Code Walkthrough	54
6.8	Visual Basic .Net Example	55
6.8.1	Building the VB.Net Example	57
6.8.2	Running the VB.Net Example	57
6.8.3	Code Walkthrough	59
Chapter 7 – 89600 VSA: Running In-Process and Sharing Hardware Driver Sessions		62
7.1	Minimum Version Requirements	62
7.2	Preparing the 89600	62
7.2.1	SCPI Preferences	63
7.2.2	Hardware Configuration	63
7.3	Starting the 89600	64
7.4	Sharing the Hardware	64
7.4.1	Checkout	64
7.4.2	Check in	65
7.5	Programming Example with 89600 VSA	65
7.5.1	C (Sharp) Example Form with 89600	65
Chapter 8 – Troubleshooting		72
8.1	Common Problems	72
8.1.1	M90XA Software fails to start when using 32-bit Applications	72
8.1.2	LTE-Advanced Apps are only available as 64-bit applications	72
8.1.3	Difficulty starting M90XA	72
8.1.4	Incorrect Alignment Results in M9393	73
8.2	Technical Support	73

Chapter 1 – Documentation Map



M90XA Startup Guide



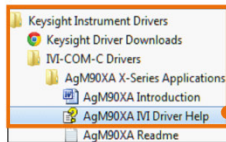
- Theory of operation & how-to procedures
- Sample programs
- Installation and configuration with the GUI & SCPI Commands
- Differences between X-Apps vs. X-Series Measurement Applications for Modular Instruments
- Troubleshooting

M90XA Revision History

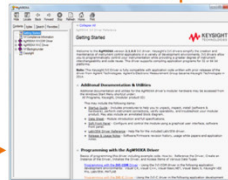


Enhancements and issues fixed in releases

Start->All Programs



M90XA IVI Driver Help

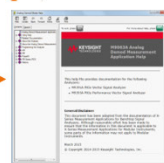


IVI-COM & IVI-C driver programmer's reference

M90XA Software GUI

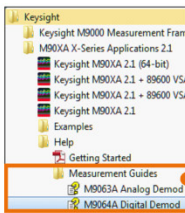


X-Series Measurement Applications Help

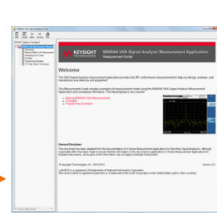


SCPI commands and queries for the basic measurement functionality

Start->All Programs

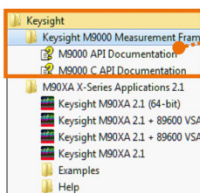


X-Series Measurement Applications Measurement Guides

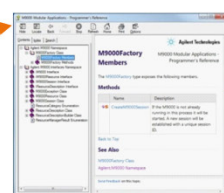


Includes examples of measurements made using the measurement application and conceptual information

Start->All Programs



M9000 API Help



Interfaces & Resources programmer's reference

M9000 C API Help



Help system for M9000 unmanaged C API

Chapter 2 – Introduction

This section provides an overview of the X-Series Measurement Applications for Modular Instruments (M90XA Software) and links to related documentation for M90XA.

- [Overview of M90XA Software](#)
- [Related Documentation](#)

2.1 Overview of M90XA Software

X-Series Measurement Applications for Modular Instruments (M90XA Software) expand the capabilities of your M9391A or M9393A PXIe Vector Signal Analyzers (PXI VSAs). These applications are the same applications used to increase the capability and functionality of Keysight's X-Series signal analyzers, and include the same GUI and SCPI programming commands. By combining the raw hardware speed of the PXI VSAs and the X-Series measurement applications, you can test more products in less time while ensuring measurement continuity from design to manufacturing. The M90XA Software transforms PXI VSAs into standards-based RF transmitter testers, providing fast RF conformance measurements to help you test your components and devices in high-volume manufacturing.

2.1.1 X-Series Measurement Applications for Modular Instruments

Following X-Series Measurement Applications are supported for Modular Instruments:

- M9063A - Analog Demodulation Measurement Application
- M9064A - VXA Vector Signal Analysis Measurement Application
- M9068A - Phase Noise Measurement Application
- M9071A - GSM/EDGE/EVO Measurement Application
- M9072A - CDMA2000/CDMAOne Measurement Application
- M9073A - W-CDMA/HSPA+ Measurement Application
- M9076A - 1xEV-DO Measurement Application
- M9077A - WLAN 802.11a/b/g/n/ac Measurement Application
- M9079A - TD-SCDMA/HSPA Measurement Application
- M9080A - LTE FDD Measurement Application
- M9080B - LTE-Advanced FDD Measurement Application

- M9081A - *Bluetooth*® Measurement Application
- M9082A - LTE TDD Measurement Application
- M9082B - LTE-Advanced TDD Measurement Application

For details on supported licenses and Product information, please refer [M90XA Software Licenses](#).

For a high level overview of the M90XA Software use model, please refer to [High Level Architecture](#).

2.2 Related Documentation

In addition to this Startup Guide, you can access the following documents relevant to M90XA Software.

Document	Description	Format	Path to Access
User's and Programmer's Guides <ul style="list-style-type: none"> • M9063A Analog Demod • M9064A VXA Signal Analyzer • M9068A Phase Noise • M9071A GSM/EDGE • M9072A cdma2000 • M9073A WCDMA • M9076A 1xEV-DO • M9077A WLAN • M9079A TD-SCDMA • M9080A LTE 	Includes SCPI commands and queries for the basic measurement functionality.	CHM	Start > All Programs > Keysight > M90XA X-Series Applications 2.1 > Help Alternatively, Launch M90XA Software, switch mode to select specific application and press F1

Document	Description	Format	Path to Access
<ul style="list-style-type: none"> • M9080B LTE-A FDD • M9081A Bluetooth • M9082A LTE-TDD • M9082B LTE-A TDD 			
Measurement Guides <ul style="list-style-type: none"> • M9063A Analog Demod • M9064A VXA Signal Analyzer • M9068A Phase Noise • M9071A GSM/EDGE • M9072A cdma2000 • M9073A WCDMA • M9076A 1xEV-DO • M9077A WLAN • M9079A TD-SCDMA • M9080A LTE • M9080B LTE-A FDD • M9081A Bluetooth • M9082A LTE-TDD • M9082B LTE-A TDD 	Includes examples of measurements made using the measurement application and conceptual information.	CHM	Start > All Programs > Keysight > M90XA X-Series Applications 2.1 > Help > Measurement Guides
M9000 API	Provides detailed documentation of	CHM	Start > All Programs >

Document	Description	Format	Path to Access
Documentation	Interfaces & Resources. Programmer's reference for the M9000 .NET API which provides the ability to share resources between different callers in the same process.		Keysight > Keysight M9000 Measurement Framework
M9000 C API Documentation	Help system for M9000 unmanaged C API.	CHM	Start > All Programs > Keysight > Keysight M9000 Measurement Framework
M90XA IVI Driver (Documentation)	Provides detailed documentation of the IVI-COM and IVI-C driver API functions, as well as information to help you get started with using the IVI drivers in your application development environment.	CHM	Start > All Programs > Keysight Instrument Drivers > IVI-COM-C Drivers > AgM90XA X-Series Applications
Modular Hardware Documentation	Provides hardware-specific documentation for the PXI modules. Includes hardware specifications and instructions on how to install, update, verify, and troubleshoot the measurement hardware.	various	Start > All Programs > Keysight > M9391 and Start > All Programs > Keysight > M9393
Revision History	Provides a list of enhancements and issues resolved in the various releases.	PDF	Start > All Programs > Keysight > M90XA X-Series Applications 2.1 > Help

- For Portable version of User documentation (PDF and CHM), go to the [product website](#) and download files from the **Document Library**.

Chapter 3 – Installation and License Setup

This section describes software pre-requisites and how to install and setup licenses for M90XA Software.

- [System Requirements](#)
- [Installing M90XA Software](#)
- [Setting up Licenses](#)

3.1 System Requirements

Following is the Keysight's recommended configuration for M90XA Software:

CPU	2 GHz or faster 64-bit (x64) processor
RAM	8 GB recommended
Hard Disk	2 GB available
Additional Drives	DVD to load software; license transfer requires network access, USB Flash drive, USB hard drive, or USB DVD drive
Operating System	Microsoft Windows® 7 Professional, Enterprise, or Ultimate (64-bit)

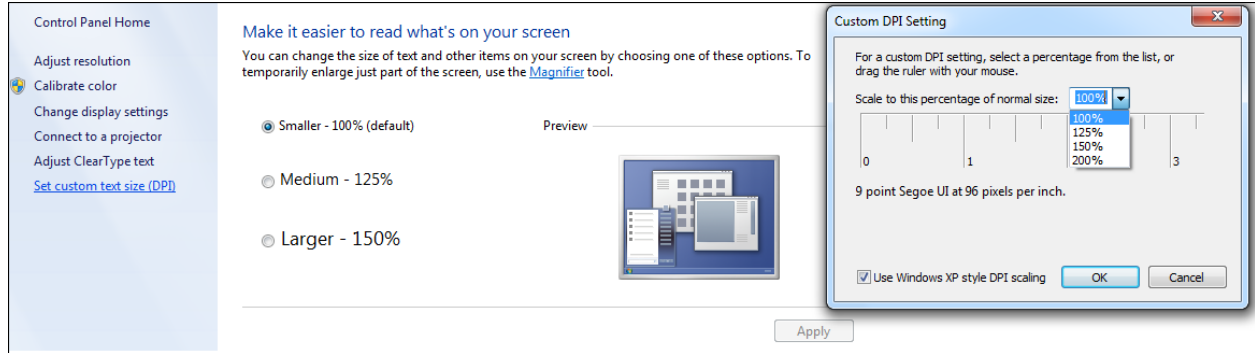
M90XA Software is no longer supported on 32-bit machines. The 32-bit version of the software, however, can be installed on top of a 64-bit machine.

3.1.1 Setup Notes

For appropriate viewing of the X-Apps for Modular Instruments, screen resolution must be set to **96dpi** or lower.

Set the custom text size (DPI) to 100%.

Go to **Control Panel > Display > Set custom text size (DPI)**. Select **100%** in the **Scale to this percentage of normal size** drop-down list.



3.2 Installing M90XA Software

This section describes the prerequisites and procedure to install the M90XA Software on your computer.

The M90XA Software must be installed on the same computer that shares the PXI backplane with the M9391A or M9393A VSAs.

3.2.1 Prerequisites

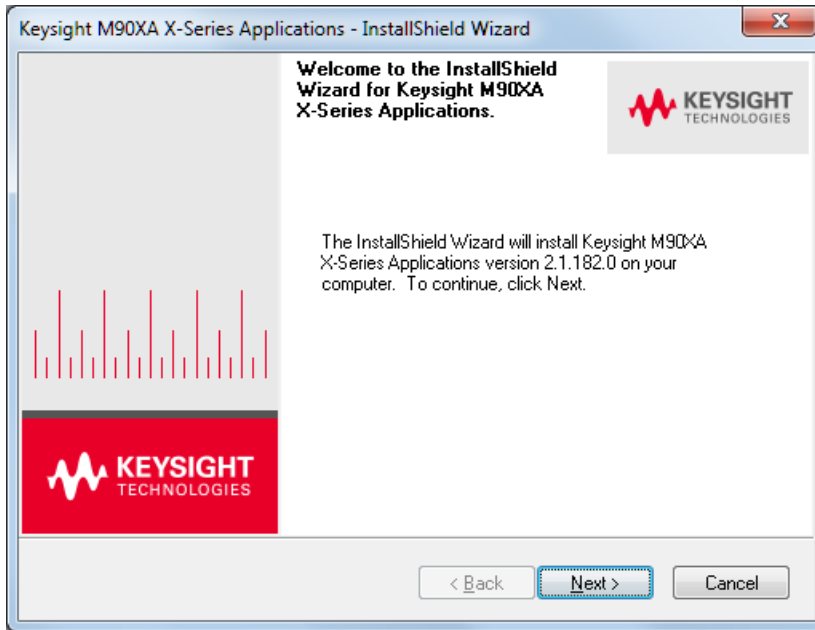
- Agilent/Keysight IO Libraries and M9391A or M9393A driver must be installed before installing M90XA Software.
 - Agilent/Keysight IO Libraries Suite: *16.3.17914.4* or higher version
 - Minimum required VSA hardware drivers:
 - M9391A driver: 1.2 or higher version (1.2.417.1)
 - M9393A driver: 1.1 or higher version (1.1.518.1)

3.2.2 M90XA Software Installer

Download the installer package from <http://www.keysight.com/find/M90XA-SW>.

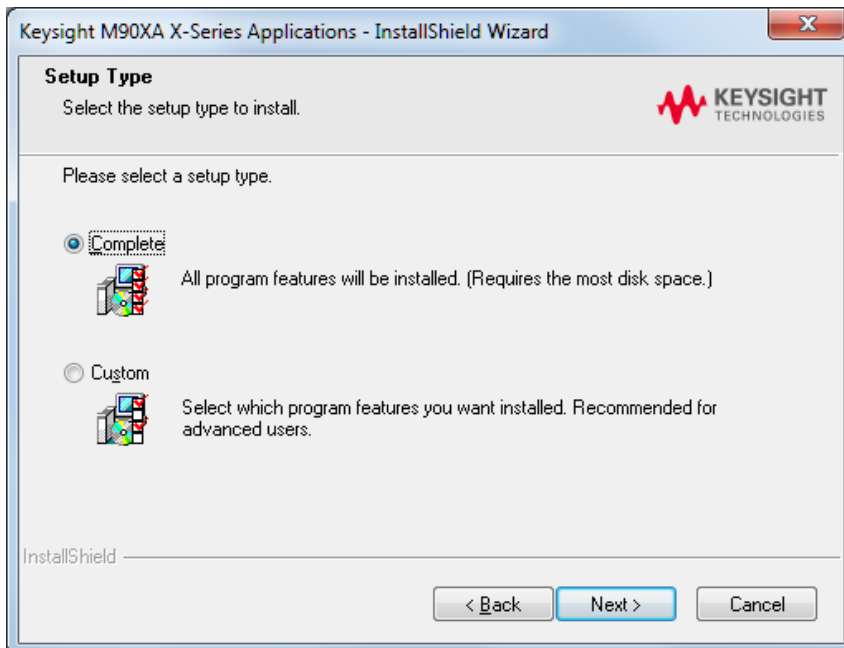
3.2.3 Installation Procedure

Launch the installer file to run the setup.



The version number of Keysight M90XA X-Series Applications in the above screen will match the version being installed.

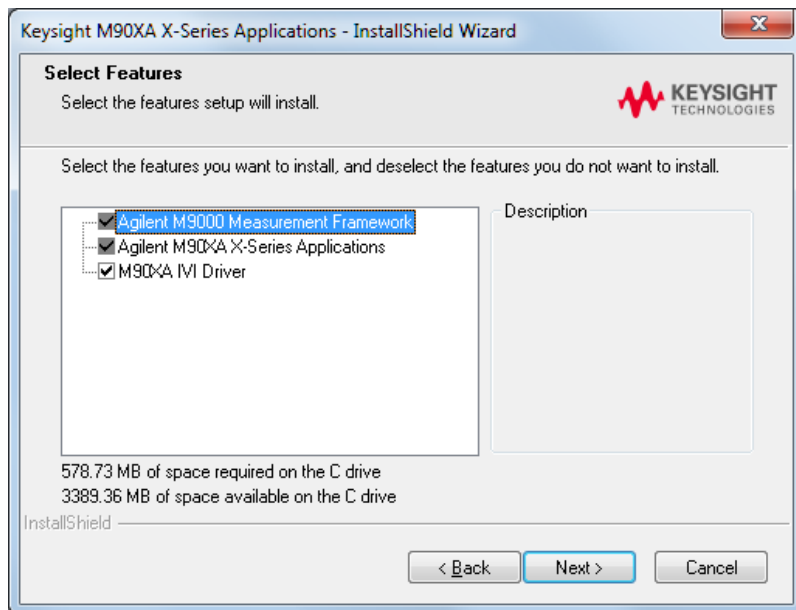
After you accept the terms of license agreement, the Installation wizard prompts you to choose either a **Complete** or **Custom** setup to install.



The Custom setup enables you to choose the features to install, out of the following:

- Agilent M9000 Measurement Framework
- Agilent M90XA X-Series Applications
- M90XA IVI Driver

This enables you to install only Agilent M9000 Core software and the M90XA IVI Driver in case you do not want to run M90XA X-Series Measurement Applications.



Follow the guided tour to complete the installation.

After the installation completes, the following applications get installed on your PC if you choose complete installation:

- Keysight M90XA X-Series Measurement Applications (This includes Agilent/Keysight M9000 Core Software)
- M90XA IVI Driver (AgM90XA)
- Keysight License Manager

3.3 Setting up Licenses

This section describes Keysight software licensing and the licenses required to run M90XA Software.

3.3.1 M90XA Software Licensing

Licenses for M90XA Software are Transportable Perpetual licenses. Perpetual licenses do not expire. Transportable licenses can be transferred from one controller or computer to another, for example from an

embedded PXI controller to an external PC. A single M90XA Software license will allow running up to four instances of the software with up to four VSA hardware sets (measurement channels) in one PXI chassis.

When you purchase M90XA Software, you are provided with a License Redemption Entitlement Certificate that documents the licenses to which you are entitled. You can redeem the certificate at <http://www.keysight.com/find/softwaremanager> using the purchase order number, entitlement certificate number, the host ID of the target computer/controller and other details as required by the redemption process. Additional instructions on how to find your host id are available at <ftp://ftp.keysight.com/pub/dsp/products/HostID.pdf>.

The Keysight Software Licensing system issues a license file for each purchased license, by e-mail, to an address you specify. Each license file is a plain text file attachment with a file name in the form XXXXXX_YYYYYY.lic. When you receive the e-mail, save the attached .lic file(s) to the Agilent license directory, which is located at: *C:\Program Files\Agilent\licensing*.

Once installed, software licenses for Keysight products can be viewed and managed from the Keysight License Manager. You can launch the Keysight License Manager either from **Start>All Programs>Keysight**



License Manager or from the system tray.

The licenses for M90XA Software must be installed by a user with System Admin privileges.

M90XA Software Licenses

The following table describes the license options for the various features of M90XA Software.

Product/Feature	Description
M9063A-2	Analog Demodulation Measurement Application
M9063A-3	FM Stereo Measurement Application
M9063A-A	Analog Demodulation Enhancements
M9064A-1	Vector Signal Analysis Application, VXA
M9064A-2	Flexible Digital Demodulation Analysis Application, VXA
M9064A-A	Vector Signal Analysis Application, VXA enhancements
M9068A-2	Phase Noise Measurement Application
M9068A-A	Phase Noise Feature Enhancements
M9071A-2	GSM/EDGE Measurement Application
M9071A-3	EDGE Evolution Measurement Application
M9071A-B	GSM/EDGE Feature Enhancements
M9071A-X	Single Acquisition Combined GSM/EDGE Measurements
M9072A-2	cdma2000 Measurement Application

Product/Feature	Description
M9073A-1	W-CDMA Measurement Application
M9073A-2	HSDPA/HSUPA Measurement Application
M9073A-3	HSPA+ Measurement Application
M9073A-D	W-CDMA Feature Enhancements
M9073A-X	Single Acq Combined W-CDMA Measurements
M9076A-1	CDMA1xEV-DO Measurement Application
M9077A-2	WLAN 802.11a/b/g Measurement Application
M9077A-3	WLAN 802.11n Measurement Application
M9077A-4	WLAN 802.11ac Measurement Application
M9079A-1	TD-SCDMA Measurement Application
M9079A-2	TD-SCDMA Measurements with HSDPA/8PSK
M9079A-A	TD-SCDMA Measurement Enhancements
M9080A-1	LTE-FDD Measurement Application
M9080B-1	LTE-FDD Measurement Application
M9080B-2	LTE-FDD Advanced Measurement Application
M9081A-2	Bluetooth Measurement Application
M9082A-1	LTE-TDD Measurement Application
M9082B-1	LTE-TDD Measurement Application
M9082B-2	LTE-TDD Advanced Measurement Application

If a license is not available or not valid, the corresponding features/options will not be enabled in the M90XA Software.

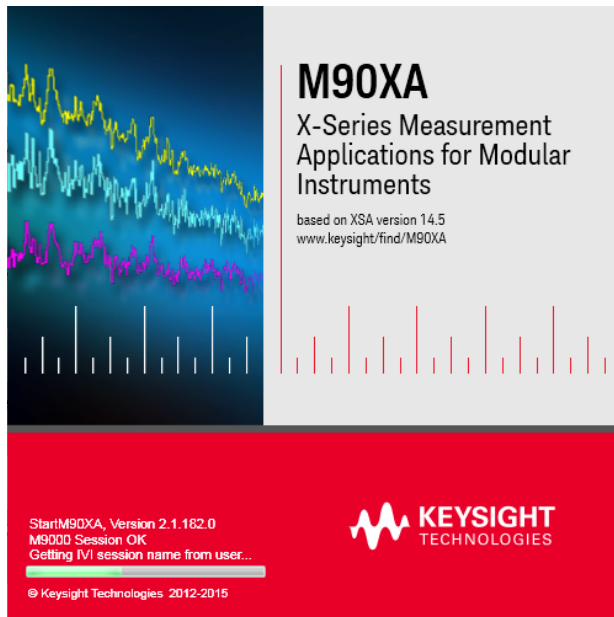
Chapter 4 – Launching M90XA

This section describes the procedure to launch the M90XA Software. It also describes the various startup options that can be used with the M90XA software.

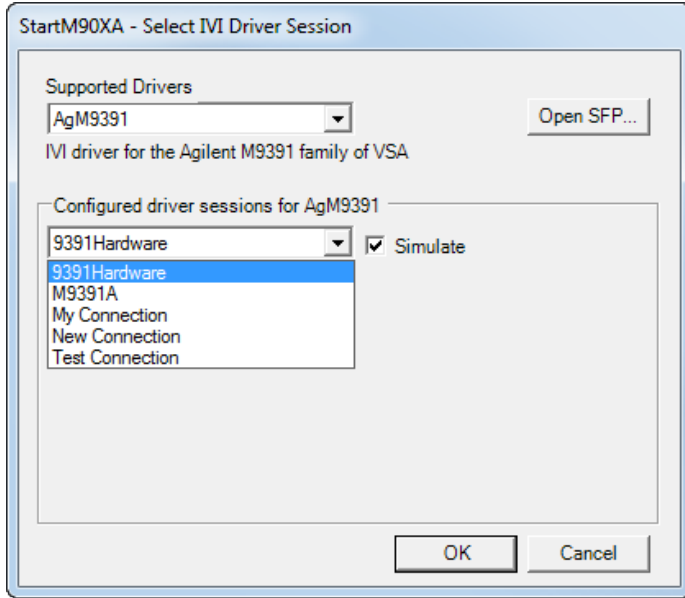
You can launch M90XA Software from the Start Menu as follows:

Start > All Programs > Keysight > M90XA X-Series Applications 2.1 > Keysight M90XA 2.1 (64-Bit).

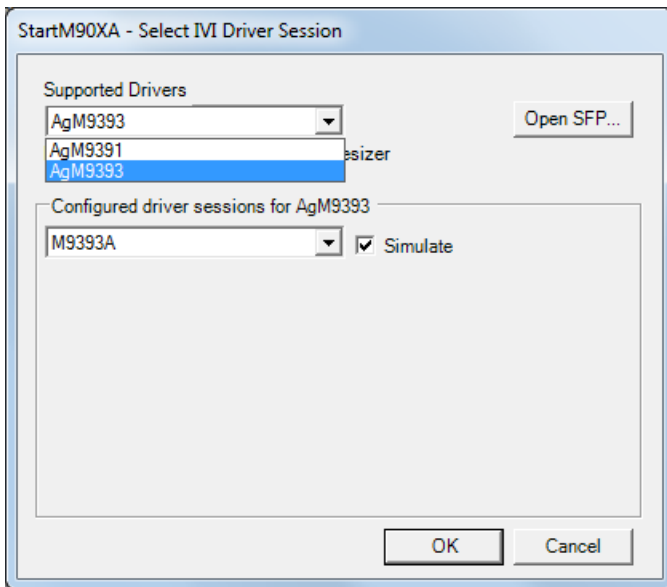
The Splash screen appears:



After a couple of seconds, the **Select IVI Driver Session** dialog appears.



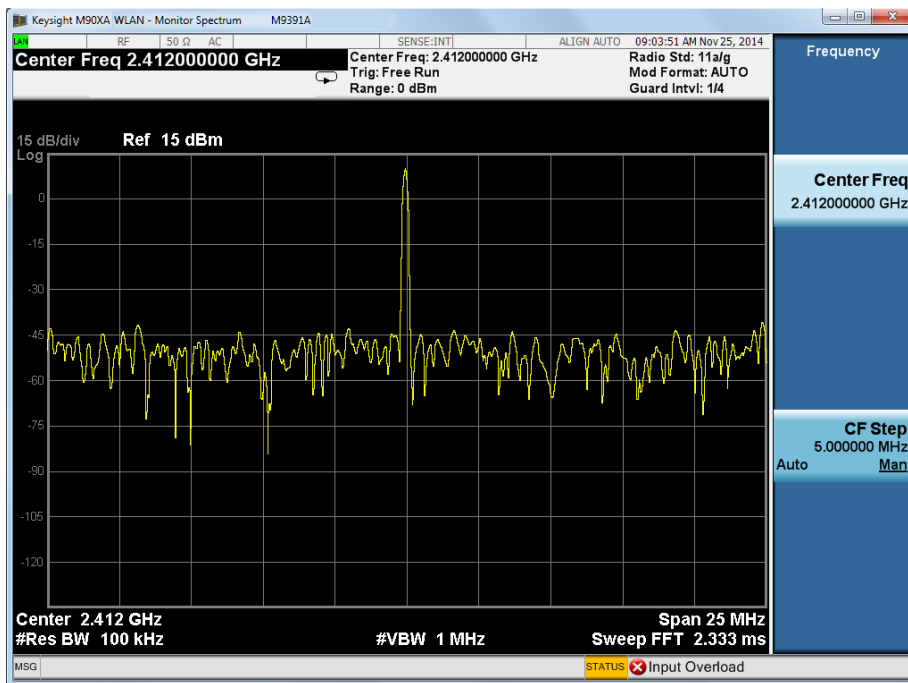
1. Select one of the two supported drivers (AgM9391 and AgM9393).



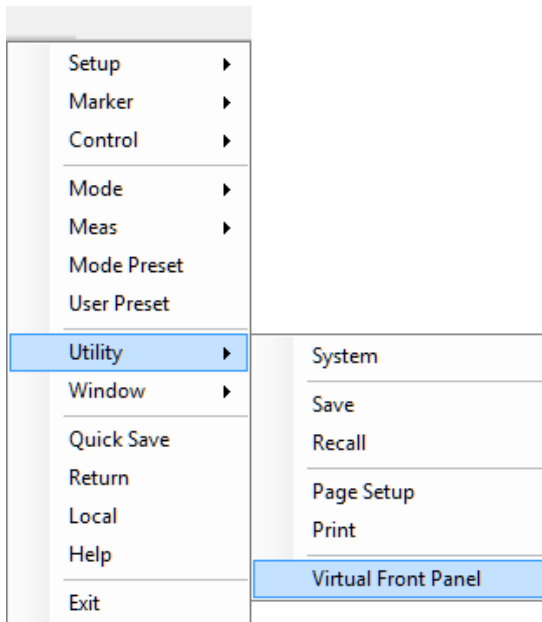
2. If you have previously configured an M9391 or M9393 driver session and saved it to the IVI Configuration Store, then you will see it in the list of **Configured driver sessions for AgM9391** or **Configured driver sessions for AgM9393**. You can skip this step and move to Step 3 directly. However, if you have not already configured an M9391 or M9393 driver session, you can configure and save it now by using the following steps (Step a through Step f).
 - a. Click **Open SFP...** to launch the Soft Front Panel.
 - b. Configure an instrument. (For details, refer to the M9391 or M9393 documentation, available at **Start>All Programs>Keysight>M9391>M9391 Help** or **Start>All Programs>Keysight>M9393>M9393 Help**.)

- c. Click **Connect** (or **Simulate**) to start driver session and the Soft Front Panel.
 - d. Click **File > Save Connection...** to save the configuration to the IVI Configuration Store by name, for example **MyVSA**.
 - e. Close the Soft Front Panel.
 - f. In the **Select IVI Driver Session** dialog, click the **Configured driver sessions for AgM9391** or **Configured driver sessions for AgM9393** drop-down list box to refresh its contents. You should now see the configuration name, for example **MyVSA**.
3. In the **Select IVI Driver Session** dialog, select the desired configured driver session.
 4. Select the **Simulate** checkbox if you want to use the M9391 or M9393 in simulation mode.
 5. Click **OK**.

This launches the M90XA Software using the selected M9391 or M9393 configuration.



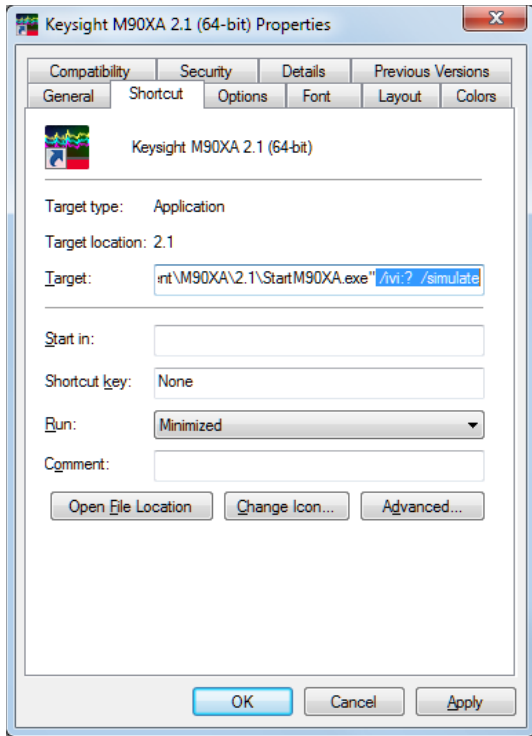
6. **Right-click** anywhere on the M90XA window to view the Context/Shortcut menu. Use this menu to access different softkey menus. Or, select Virtual Front Panel to open a second window with front panel buttons.



To start the M90XA and 89600 applications from the Start menu, select **All Programs > Keysight > M90XA X-Series Applications 2.1 > Keysight M90XA 2.1 + 89600 VSA (64-bit)** or **All Programs > Keysight > M90XA X-Series Applications 2.1 > Keysight M90XA 2.1 + 89600 VSA**.

4.1 Startup Options

There are several startup options that can be used when launching the M90XA Software. The startup options can be added to the shortcut properties at the end of the **Target** line. To open the shortcut properties window, right-click on the shortcut and select **Properties**.



Startup Option	Parameter	Description
/sim		Start in simulate mode
/ivi:?		Interactive selection of logical name of receiver hardware
/ivi:	Logical name of receiver hardware	Use specific hardware
/modes:	Comma-separated list of mode names (e.g. /modes:WLAN,EDGE GSM) Valid modes are ADEMOM, BT, CDMA1XEV, CDMA2K, EDGE GSM, LTE, LTETDD, LTEAFDD, LTEATDD, PNOISE, TDSCDMA, VSA, WCDMA, WLAN	Load specific measurement applications
/hislip:	Device number	Use a specific HiSLIP device number
/telnet:	Port	Use a specific telnet port number
/socket:	Port	Use a specific socket port number
/silent		Non-interactive mode
/debug		Leave the StartM90XA.exe console window open

Startup Option	Parameter	Description
/u or /?		Show the command line usage instructions

If you are combining multiple startup options they should be separated by spaces. There should be no spaces within a startup option.

For example, `/ivi:MyVsa /modes:WLAN,EDGE GSM /hislip:1`

Chapter 5 – Using M90XA Software

When you combine the raw hardware speeds of the PXI VSAs and the X-Series Measurement Applications for Modular Instruments, you can test more products in less time while ensuring measurement continuity from design to manufacturing.

The X-Series Measurement Applications for Modular Instruments are the same as those available for Keysight's X-Series signal analyzers. Consistent and intuitive user interfaces reduce the need for re-training or familiarization and also simplify measurement setup. Programming consistency includes SCPI compatibility across platforms to enable software re-use from benchtop to modular and speed up test system development. This consistency reduces risk in the transition from development phase to manufacturing.

In addition, Keysight's X-Series Measurement Applications for Modular Instruments include a unique resource manager that provides direct access to PXI VSA hardware drivers for the fastest power and spectrum-based measurements while simultaneously using fast modulation quality measurements provided in the X-Series Measurement Applications. Because of the resource manager, switching time between different applications is significantly less.

This section describes the following key aspects of using the M90XA software.

- [High Level Architecture](#)
- [Using the GUI](#)
- [Using SCPI](#)
- [Using a Modular VSA versus a Swept Analyzer](#)
- [Triggering and Gating](#)
- [Alignments](#)
- [Event Logging](#)
- [Optimizing Performance](#)
- [Additional Information](#)

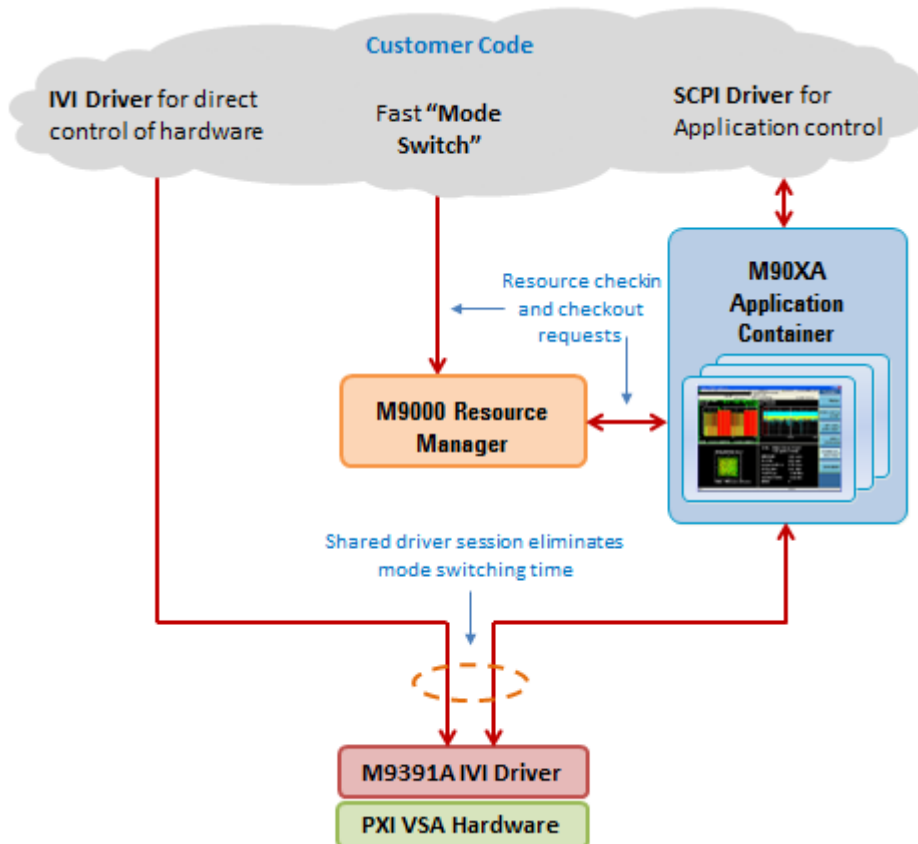
M90XA Help

Help for M90XA is available from the Start Menu or by pressing F1 while the application is running. M90XA Help is generated from common files shared with X-Series Applications because M90XA Software has been created from the same code base as X-Series Applications. Not all M90XA Help files have been updated to include specific information related to Modular X-Apps or PXI VSA hardware.

5.1 High Level Architecture

This section provides an overview of the high level architecture of the M90XA software.

The following diagram provides high level overview of the M90XA Software use model:



5.1.1 M90XA Application Container

The Keysight M90XA is a container for the X-Series Measurement Applications. M90XA provides features and services which are common to all of the measurement applications. For example, the graphical user interface, programming API, utility, and housekeeping functions (such as save, recall, system info, and error reporting). The AgM90XA is an IVI driver (IVI-COM or IVI-C), however, once the driver is initialized the measurement applications use a SCPI programming API and offers code compatibility with benchtop analyzers.

You can launch the M90XA Software from the Start menu as described in the [Launching M90XA](#) section. Once launched, make measurements in either of the following ways:

- Interactively by using the M90XA GUI
- Programmatically by using the SCPI commands

5.1.2 M9000 Resource Manager

The M9000 Resource Manager allows hardware driver session to be shared by multiple applications running in the same process without having to close and reopen a hardware driver session. The supported applications include the M90XA Software, 89600 VSA, and customer code. This capability enables high speed operations including Adjacent Channel Power Ratio (ACPR) measurement and power servo routines. In addition, it allows to use the M90XA Software for measurements that are not available in the driver such as Error Vector Magnitude (EVM) and Spectral Emission Mask (SEM).

The M9000 Resource Manager provides arbitration between multiple users based on a co-operative sharing model. Each time an X-Series Measurement Application needs to access the hardware driver session, it sends the resource manager a checkout request, uses the hardware, then immediately checks it back into the resource manager. Similarly, when you want to use the hardware driver session you must first check it out from the resource manager. For information related to how the 89600 VSA accesses the hardware driver session, refer to the [89600 VSA: Running In-Process and Sharing Hardware Driver Sessions](#) section. To programmatically access the hardware driver session refer to the [Programming with M90XA](#) section.

The M9000 Resource Manager has:

- a .NET API
- an unmanaged C API

5.1.3 Supported VSA Hardware

The supported VSA hardware for M90XA software is M9391A PXIe Vector Signal Analyzer and M9393A PXIe Performance Vector Signal Analyzer. For more details on M9391A and M9393A, refer to <http://www.keysight.com/find/M9391A> and <http://www.keysight.com/find/M9393A>, respectively.

5.1.4 M9391A and M9393A IVI Drivers

The M90XA Software connects to VSA hardware through the M9391A or M9393A IVI driver (IVI-COM or IVI-C), which is shipped with the VSA hardware. These drivers can also be download from the links below:

- M9391A Drivers - <http://www.keysight.com/find/M9391A-Driver>
- M9393A Drivers - <http://www.keysight.com/find/M9393A-Driver>

5.2 Using the GUI

Use of the M90XA PC Application is essentially the same as using remote desktop to interact with a benchtop analyzer's embedded X-Apps software. Right click anywhere on the application window to bring up menus. A Virtual Front Panel is available under the **Utility** menu. The familiar X-Series analyzer softkeys are included on the PC Application.

Some menus that were applicable only to a benchtop analyzer do not appear in the M90XA software.

5.3 Using SCPI

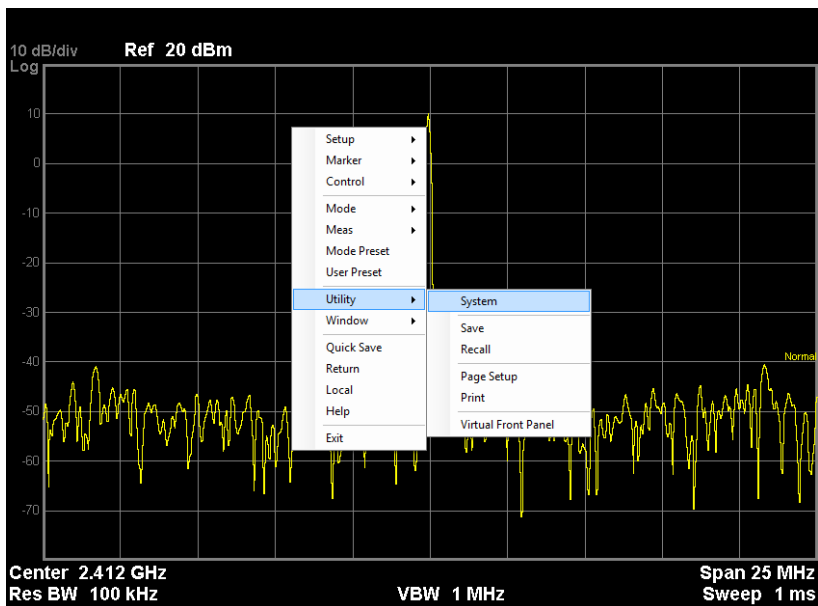
This section describes the SCPI usage of the M90XA Software.

5.3.1 SCPI Control of M90XA

Once you have launched M90XA, you can program the measurement applications by using the SCPI commands.

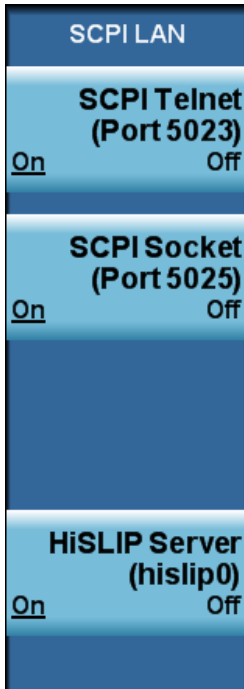
To view the SCPI I/O configuration:

1. Launch the M90XA Software (Refer Launching M90XA Software).
2. **Right-click** to view the M90XA Context/Shortcut menu.
3. In the Context/Shortcut menu, select **Utility > System**.



4. Click **I/O Config**.
5. Click **SCPI LAN**.

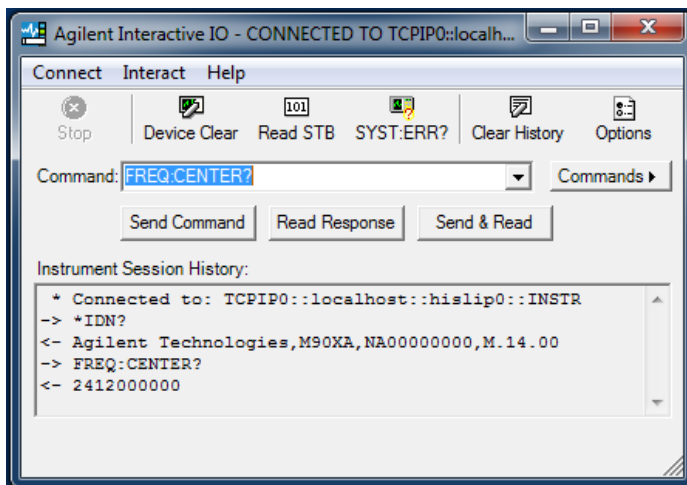
The SCPI I/O configuration appears:



Use the displayed I/O configuration information plus the IP address or host name of the computer running the M90XA software to determine the VISA resource address for this instance of the M90XA software. For example, if the M90XA shown above is running on a computer with IP address 192.168.1.1, the full SCPI Socket address is "TCPIP0::192.168.1.1::5025::SOCKET", and the full HiSLIP address is "TCPIP0::192.168.1.1::hislip0::INSTR". Use this resource address to open a VISA session for sending SCPI commands to the M90XA.

The M90XA supports SCPI Telnet, SCPI Socket, and HiSLIP connections, but not SIDL Server (VXI-11) connections.

A tool like Agilent Interactive IO (part of Agilent Connection Expert) can be used to test SCPI control. After connecting, the M90XA should respond to "*IDN?" and "FREQ:CENTER?" commands.



For details on measurements to be made for individual X-Series measurement application, you can refer to the respective documentation available at **Start Menu>All Programs>Keysight>M90XA X-Series Applications 2.1>Help**.

5.3.2 HiSLIP

The M90XA supports SCPI connections using HiSLIP, Socket or Telnet protocols.

HiSLIP (High-Speed LAN Instrument Protocol) is recommended because it is fast, it supports instrument control features such as Device Clear and Service Request, and it supports multiple instruments at a single IP address.

HiSLIP addresses include a HiSLIP device name like "hislip0" ending in a HiSLIP number. By default the M90XA selects the next available number at startup. If two instances of the M90XA are started, the second instance will usually be at hislip1. When using a shortcut to StartM90XA.exe to start the M90XA, this address uncertainty can be avoided by adding the /hislip command line option with a large number such as 101. When using the AgM90XA IVI driver to start the M90XA from a program, the assigned address can be queried by reading the HiSLIPVisaAddress property.

The M90XA SCPI Socket Server must remain On. Do not turn SCPI Socket Server Off because StartM90XA (version 2.0 and later) requires SCPI Socket Server to complete the startup. M90XA versions prior to 2.0 require HiSLIP Server On to complete startup.

5.3.3 SICL Server (VXI-11)

As noted earlier, M90XA supports SCPI connections using HiSLIP, Socket, or Telnet protocols. Unlike benchtop instruments, it does not support the SICL Server (VXI-11) protocol. Not supporting SICL Server enables the M90XA to run on a computer with other instrument applications.

The recommended TCPIP protocol for M90XA is High-Speed LAN Instrument Protocol (HiSLIP). For more information about using HiSLIP, see "Using the TCPIP Interface Type for LAN Access" in the *Keysight IO Libraries Connectivity Guide*.

SICL Server connections are typically established with a VISA resource string ending in **::instr0::INSTR**. Controlling programs can be changed to use HiSLIP by changing the VISA resource string to end in **::hislip0::INSTR** (for example).

5.4 Using a Modular VSA Versus a Swept Analyzer

This section describes some of the differences when using X-Series Applications for Modular Instruments versus using swept-tuned analyzers, as in case of Benchtop instruments.

- **Power Range**
- **Spectrum Measurements**
 - **Stepped FFT vs Swept-tuned for Spectrum Measurements**
 - **Image Protection**

5.4.1 Power Range

X-Series benchtop analyzers (MXA, PXA) typically have an amplitude (Y Scale) setup menu that controls the analyzer's internal attenuators, preamp, and IF Gain. The amplitude level is controlled using RF attenuation, preamp, and IF gain commands as shown below:

```
:POWer[:RF]:ATTenuation <atten>
:POWer[:RF]:EATTenuation <atten>
:POWer[:RF]:GAIN[:STATe] OFF|ON
:IF:GAIN:FFT[:STATe] AUTOrange|LOW|HIGH
```

M90XA with modular PXI VSA hardware has an amplitude setup menu with a range entry. Select the peak power level expected to be present at the PXI VSA input and the application will optimize the internal attenuators for that expected input level. On the X-Series Measurement Applications for Modular Instruments, amplitude level is controlled using the Range command:

```
:POWer:RANGe <dBm>
```

This command sets the expected input power (peak) in dBm.

Sending Attenuator commands to the M90XA will cause an "Undefined header" error, and have no effect on amplitude. Most applications can be updated by mapping 0 dB attenuation to -10 dBm Range, 20 dB attenuation to +10 dBm Range, and so on.

5.4.2 Spectrum Measurements

Stepped FFT vs Swept-tuned for Spectrum Measurements

- X-Apps on Keysight X-Series benchtop analyzers (MXA, PXA) utilize the instrument's Spectrum Analyzer architecture:
 - Swept-tuned data acquisitions (Spectrum measurements) operate by sweeping an LO through the frequency range of interest, acquiring spectrum data as it sweeps.
 - Time data acquisitions (IQ measurements) set the LO to a fixed frequency and acquire time-record data.

- X-Apps on Keysight Modular PXI Instruments utilize the modular instrument's architecture to acquire spectrum and time data. In the case of a VSA (Vector Signal Analyzer) instrument:
 - Time data acquisitions (IQ measurements) use a fixed LO, the same as in a Spectrum Analyzer.
 - Spectrum data acquisitions (Spectrum measurements) are acquired by collecting and converting time data to spectrum data using FFT techniques. If more spectrum data is needed than the acquisition bandwidth supports, multiple FFTs are acquired and stitched together to form the wider span spectrum.
- For Spectrum Measurements on Keysight Modular PXI Instruments, Sweep Time is interpreted as “Equivalent Sweep Time”, such that the measurement results will have equivalent variance within an RBW as that of a measurement made using swept hardware. Actual acquisition times are typically much shorter than the Equivalent Sweep Time value. Use Gate Length if a specific acquisition time is needed.

Image Protection

Modular instrumentation uses different frequency downconversion techniques, resulting in some differences in mixer image protection.

Analyzers use super heterodyne techniques, where input signals are shifted to lower frequencies using one or more mixers. Without any filtering or image suppression signal processing, the analyzer will respond at both the desired frequency and also at an image frequency. The image-susceptible frequency range is typically 400 MHz to 800 MHz above or below the analyzer center frequency. To optimize measurement speed, modular analyzers use different image protection techniques.

- The X-Series benchtop analyzers (MXA, PXA) utilize fixed filtering below 3.6 GHz, and a YIG-tuned preselection filter above 3.6 GHz.
 - Below 3.6 GHz all measurements are image protected.
 - Above 3.6 GHz measurements are image protected only when the RF preselector is on. There is no image protection when the RF preselector is bypassed. Modulation accuracy usually bypass the preselector and so are susceptible to images.
- The M9391A Modular PXI analyzer can be configured to use single mixer conversion mode without image protection, or image-protected double conversion mode. For best EVM and phase noise performance, the X-apps use the single mixer conversion mode for all measurements except Spurious Emissions. Spurious Emissions uses image protected double downconversion mode.
- The M9393A Modular PXI microwave analyzer utilizes fixed filtering below 3.6 GHz, and may be configured to use image suppression signal processing above 3.6 GHz.
 - Below 3.6 GHz all X-app measurements are image protected.
 - Above 3.6 GHz X-app measurements are not image protected, with the exception of Spurious Emissions (all modes) and the WLAN 80+80 Spectrum Emission Mask measurement.

- The Spurious Emissions measurements and the WLAN 80+80 Spectrum Emissions Mask measurement are always image protected using image suppression signal processing. This technique performs multiple acquisitions at each frequency and is described in application note *Achieving Excellent Spectrum Analysis Results Using Innovative Noise, Image and Spur-Suppression Techniques* <http://literature.cdn.keysight.com/litweb/pdf/5991-4039EN.pdf>. Because multiple acquisitions are combined (see Figure 7 of the note), this technique requires the use of gating when measuring burst signals, and requires a low video bandwidth value to reduce the variance of time-varying signals.

When making measurements without image protection, insure the signal at the analyzer input does not include power in the image-susceptible frequency range.

The Monitor Spectrum measurements may not be image protected. One purpose of Monitor Spectrum is to show the spectrum of the input signal under the same setup as other measurements like Modulation Accuracy. Contact Keysight support if your application requires Monitor Spectrum measurements with image protection.

5.5 Triggering and Gating

Since the PXI VSA trigger hardware is not identical to the trigger hardware in the X-Series benchtop analyzers, and since the PXI VSA is Fixed or Stepped FFT and not Swept-tuned for spectrum measurements, trigger and gating capabilities differ according to the capabilities of the underlying hardware. The following section provides more information.

5.5.1 Trigger Differences

This section lists the trigger capability differences between X-Series Measurement Applications for Benchtop Analyzers vs. X-Series Measurement Applications for Modular Instruments. The following table compares the X-Series benchtop analyzer trigger sources to modular instrument trigger sources.

Trigger Source	X-Series Benchtop Analyzer	M9391A and M9393A VSA Modular Instruments
Free Run	Supported	Supported
Video (IF Envelope)	IF signal level	IF signal Level; uses the driver's Magnitude Trigger mode
Line	Power line	Not available
External 1	"Trigger 1 In" on rear panel	"Trig 1" on M9214A IF Digitizer PXI module
External 2	"Trigger 2 In" on rear panel	Chassis backplane PXI TRIG 2 line
RF Burst	RF (wideband)	Uses the M9391A Wideband Magnitude Trigger mode with 80 MHz of

Trigger Source	X-Series Benchtop Analyzer	M9391A and M9393A VSA Modular Instruments
	signal level	trigger bandwidth, or the M9393A Wideband Burst Detector with at least 200 MHz of trigger bandwidth.
Periodic Timer (Frame)	Supported	Supported (see note below)

External 1 Trigger

"External 1" uses the M9214A Digitizer module "Trig 1" port as an external trigger input. Level, slope, and delay can be set. This input is not configured to provide a 50 ohm termination.

External 2 Trigger

"External 2" uses a PXI chassis backplane trigger as an external trigger input. The specific trigger line used is TTL TRIG 2. Slope and delay can be set. Level is ignored. To use this in a PXI system, configure another PXI module to drive TTL TRIG 2. The chassis trigger bus routing must also be configured if the modules are located in different chassis trigger bus segments.

RF Burst Trigger

"RF Burst" uses the M9391A VSA Wideband Magnitude trigger mode which can be used to trigger from a burst signal when tuned to measure in an adjacent frequency channel. With M9391A VSA the bandwidth of this trigger detector is 80 MHz, which is less than the RF Burst trigger bandwidth of most X-series Benchtop analyzers.

With M9393A VSA the bandwidth of this trigger is at least 200 MHz which is the same as most X-series Benchtop analyzers.

Operation of the M9393A burst trigger requires the use of a PXI backplane trigger line (usually PXI_TRIG 5), from the M9365A Downconverter module (where the detector is located) to the M9214A Digitizer module. If these modules are located in different PXI backplane trigger bus segments, a trigger route must be configured. When running in a M9018A chassis the M90XA software will automatically configure trigger routing when the software is started. If the system requires use of different trigger line, the line can be specified in the Advanced Options of the stored hardware configuration (or alias). For example, "BurstTriggerLine=5" will use PXI_TRIG 5.

Periodic Timer (Frame) Trigger

Periodic Timer (Frame) trigger is supported, with differences in synchronization behavior. On modular hardware, if Sync Source is set to a trigger source, the next measurement will always wait for that trigger event to re-synchronize before completing the measurement. On X-series Benchtop analyzers in this same setup, measurements do not wait to re-synchronizing if the Periodic Timer trigger was previously synchronized. This can be noticeable when synchronizing to a 1 pulse-per-second external trigger. It can be avoided by setting Sync Source to Off after synchronizing the first measurement.

Gating

Many X-series applications have support for Gating under the Sweep/Control key. This is important when measuring burst signals and when the measurement time is longer than the duration of a single burst. In a traditional swept-tuned analyzer, gating suspends the sweep between bursts. In a fixed-tuned modular analyzer, gating controls the trigger source and acquisition length of individual acquisitions. It is per-acquisition triggering. The Gate Source can be External 1, External 2, RF Burst, or Periodic Timer. If Gate is On, but a valid gate/trigger is not present, the analyzer will display "Waiting for Trigger". If you see "Waiting for Trigger", and the analyzer trigger is configured for Free Run, remember to check the Gate settings.

Trigger Outputs

When using the M9391A or M9393A VSA modular hardware, the following trigger outputs are available. These outputs produce a 10 μ s TTL pulse when the event occurs. A single measurement may perform multiple acquisitions and cause multiple pulses to be output.

Trigger Output	Function	M9391A VSA Connector Location	M9393A VSA Connector Location
Armed	Pulses when the analyzer is armed and ready to receive a trigger.	Trig 2 on M9350A Downconverter module	Trig 2 on M9365A Downconverter module
Trigger Received	Pulses when the analyzer has started an acquisition.	Trig 2 on M9214A IF Digitizer module	Trig 2 on M9214A IF Digitizer module
Acquisition Complete	Pulses when the acquisition is complete.	Trig 2 on M9301A Synthesizer module	Trig 2 on M9308A Synthesizer module

5.6 Alignments

This section provides information about M90XA alignments.

5.6.1 Turning Off Auto Alignments

Since alignments of M9391A or M9393A PXIe Vector Signal Analyzers can require anywhere from several seconds to a few minutes to complete, it is recommended that for automated test applications, auto alignments be turned off.

The SCPI command to turn off auto alignments is:

```
CAL:AUTO OFF
```

For M90XA, CAL:AUTO PART has the same behavior as CAL:AUTO OFF.

Automated test applications can use the calibration expired query to check for & perform an alignment as needed:

CAL:EXP?

This command is a blocking query; thus, it requires a VISA timeout that is longer than the longest alignment time.

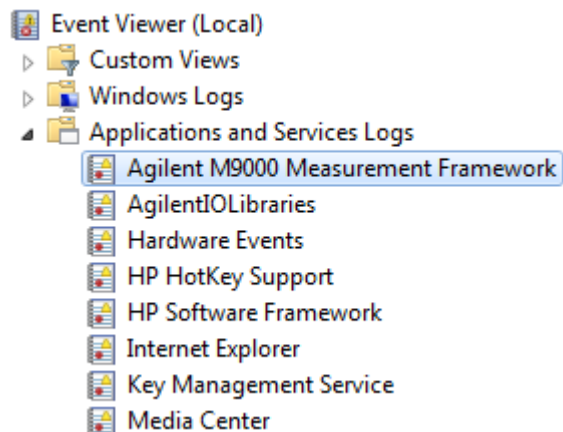
5.6.2 M9393A Alignments

The M9393A PXIe Vector Signal Analyzer alignment requires that no signal be present at the analyzer's RF input. When an alignment is attempted, M90XA performs a check for power at the input. If a signal is detected, the alignment will be skipped and an error "Align Now; RF Required" will be posted to the M90XA GUI and Error Log. The M90XA Error Log and the M9000 Event Log will also report that the alignment was skipped. Automated test applications should ensure no power is applied to the M9393A RF input before sending CAL:EXP?

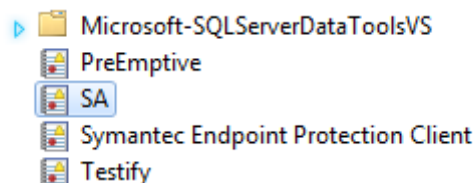
5.7 Event Logging

The event logging information is available as entries in the Windows Event Logger

Events from the M9000 Resource Manager are logged under **Applications and Services Logs -> Agilent M9000 Measurement Framework**



Events from the M90XA X-series Measurement Applications are logged under **Applications and Services Logs -> SA**



5.8 Optimizing Performance

This section provides techniques for optimizing performance of M90XA.

- PXIe Chassis Configuration
- Controller Configuration

5.8.1 PXIe Chassis Configuration

Depending on your model of chassis and type and model of controller, you might be able to achieve additional measurement speed by modifying the PCIe Link Configuration of the chassis. Please see your chassis User Guide for more information.

5.8.2 Controller Configuration

Windows Operating System

You may be able to improve performance by optimizing settings under the Windows Control Panel.

- **Power Options:** Verify that a High Performance Plan is selected and that Processor Power Management is set to 100% for Minimum and Maximum Processor State.

Controller BIOS

Your controller's BIOS may have CPU settings that favor efficient use of power over maximizing performance. You might see performance improvements after disabling these energy conservation features.

5.9 Additional Information

This section describes additional items to be aware of when using M90XA with the M9391A or M9393A PXIe Vector Signal Analyzer:

5.9.1 M9063A Analog Demod

- The "Demod to Speaker" feature that enables a demodulated audio signal to connect to the speaker is not supported.

Chapter 6 – Programming with M90XA

The M90XA Software includes several example programs that illustrate the SCPI usage of the M90XA Software and the shared usage of the M9391 or M9393 IVI Driver Session. Please refer to the [Using SCPI](#) section for basic information related to SCPI usage.

6.1 M90XA Example Programs

Each of the example programs show you how to perform the following tasks:

- Open an M9000 Resource Manager session.
- Create a sharable M9391 or M9393 IVI driver session.
- Start the M90XA Software.
- Quickly switch between the X-Series Measurement Applications and direct access to the driver by sharing the IVI driver session.

The example programs can be accessed via **Start Menu > All Programs > Keysight > M90XA X-Series Applications 2.1 > Examples > Open Examples Folder**. Following are the example folders:

Folder Name	Sub-folder name	Language	M9391	M9393	Description
LabVIEW	LabVIEW 2011	LabVIEW	•		LabVIEW Example
LabVIEW	LabVIEW 2013	LabVIEW	•		LabVIEW Example
MATLAB	-	MATLAB	•	•	MATLAB Example
VS.Net	AnsiC > C_UsingM9391	C/C++	•		C or C++ Example
VS.Net	AnsiC > C_UsingM9393	C/C++		•	C or C++ Example
VS.Net	CSharp > CS_ExampleForm	.NET (C#)	•	•	C (Sharp) Example Form
VS.Net	CSharp > CS_ExampleFormWith89600	.NET (C#)	•	•	C (Sharp) Example Form with 89600
VS.Net	CSharp > CS_UsingM9391	.NET (C#)	•		C (Sharp) Example
VS.Net	CSharp > CS_UsingM9393	.NET (C#)		•	C (Sharp) Example
VS.Net	VB > VB_UsingM9391	Visual Basic .Net	•		Visual Basic .Net Example
VS.Net	VB > VB_UsingM9393	Visual Basic .Net		•	Visual Basic .Net Example

1. Before compiling an example program, you should first copy the example folder to another location that you will use for development. This will ensure that you always have access to the original examples.
2. Before running an example program, you should be able to manually launch the M90XA Software using a configured M9391 or M9393 driver session name (refer Launching M90XA Software). This configured driver session name (or alias) is used in the .NET and LabVIEW example programs to programmatically open the M9391A or M9393A driver session.

- For programming documentation on the M9000 Resource Manager's .NET API and unmanaged C API, please refer to their respective documentation available at **Start Menu>All Programs>Keysight>Keysight M9000 Measurement Framework**.
- For programming documentation on the M90XA IVI drivers, M9391 IVI drivers, and M9393 IVI drivers, please refer to their respective documentation available at **Start Menu>All Programs>Keysight Instrument Drivers**.

6.2 Shared Native .dlls

When running one or more applications in a single process space, it is important that the loaded version of a shared native (unmanaged) .dll be one that meets the requirements of all the applications in the process. Without intervention, the first application to load a .dll may load a version that is older than what is needed by other applications in the process.

To address this need, the M9000 loads shared native .dlls when a M9000 session is opened. The M9000 loads the shared native .dlls from the folder(s):

- 64-bit application: *C:\Program Files\Agilent\M9000\2.1\Core\CommonFiles*
- 32-bit application: *C:\Program Files (x86)\Agilent\M9000\2.1\Core\CommonFiles*

Usage Notes:

- You can put newer versions of shared .dlls in this folder if needed.
- A “repair” of the M90XA install will restore the .dlls to the original versions that shipped with M90XA.
- Any issues with loading shared .dlls will be reported in the M9000 Event Log.
- If a .dll in this folder is newer than one used by another application in your process, and that application must start before the M9000 session is opened, you may copy the newer .dll to your application’s folder (from which the older version .dll is being loaded).

6.3 LabVIEW Example

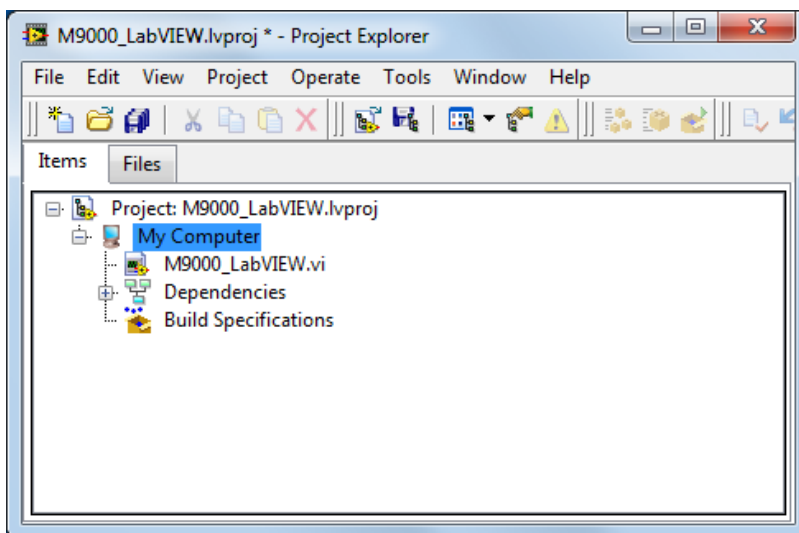
In the **LabVIEW** folder available in the Examples directory, you can find the example code for the 32-bit version of National Instruments' LabVIEW (versions 2011 and 2013). Copy the example folder to another location that you will use for development. This will ensure that you always have access to the original examples.

6.3.1 Prerequisites

Before you execute the examples, it is required that you switch the .NET Common Language Runtime (CLR) for LabVIEW to version 4.0. This is not a standard configuration for LabVIEW. You can refer to the detailed instructions for this at the National Instruments' web site - http://zone.ni.com/reference/en-XX/help/371361J-01/lvhowto/configuring_clr_version.

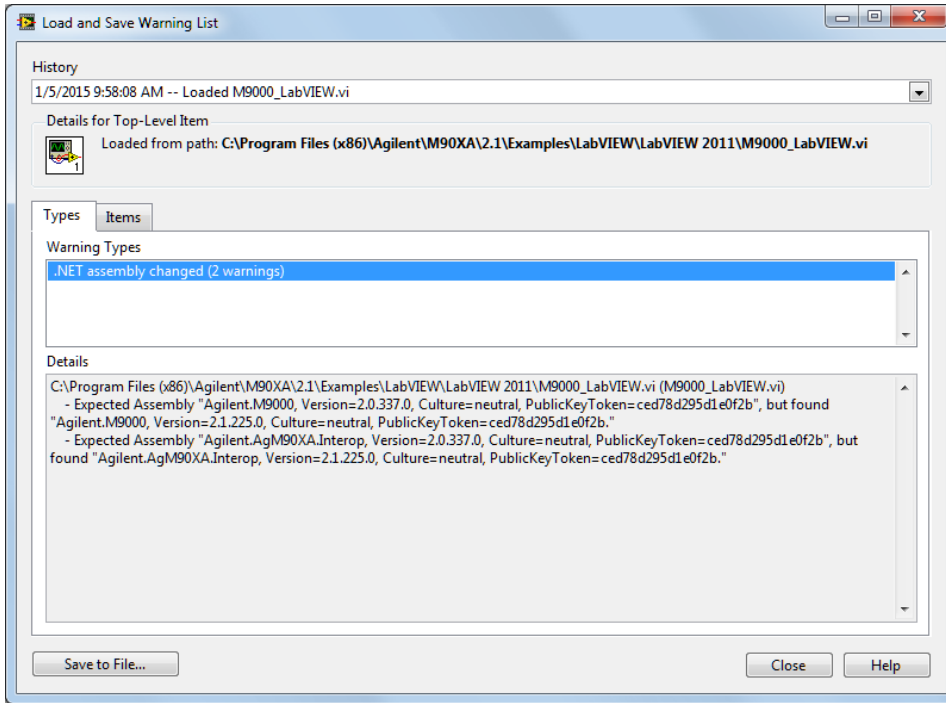
6.3.2 Executing the LabVIEW Example

1. Double-click the **M9000_LabVIEW.lvproj** file available from the location where you copied the example code. The Project Explorer page appears:



2. Double-click the **M9000_LabVIEW.vi** file on the Project Explorer page to open the VI. The Front Panel appears.

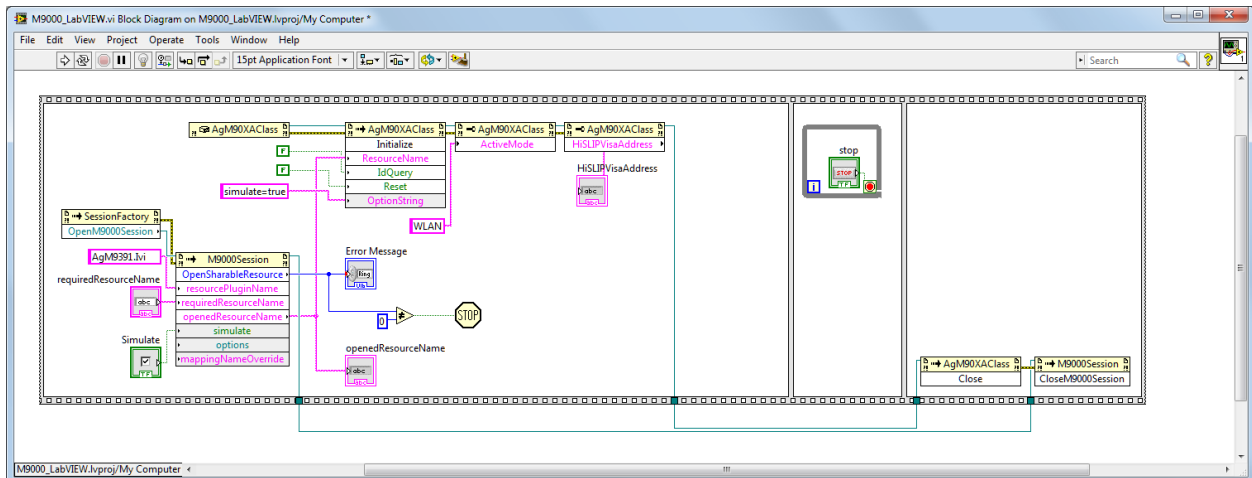
Two warnings may appear (as displayed below) about the changed .NET assembly versions. You may ignore these warnings and click **Close**.



3. The front panel of the M9000_LabVIEW.vi has a single input field called *requiredResourceName*. Enter the resource configuration name you set up earlier (for example MyVSA, or M9391), during launching of M90XA, in the *requiredResourceName* field. Check the Simulate box if you want to simulate hardware.
4. Click the **Run** button to run VI. This launches the GUI of the M90XA Software.
5. To close the M90XA GUI, click the STOP button on the front panel of the M9000_LabVIEW.vi. This will also stop the LabVIEW example.

6.3.3 The LabVIEW Example Code

When you access the Block Diagram by using the **M9000_LabVIEW.vi Front Panel**, you can view the example code:



This example code consists of a sequence of **Invoke Nodes** that call methods on the AgM90XA IVI driver as well as the M9000 API.

Please refer to the respective API documentation for more detail.

SCPI strings can be used to make measurements using the M90XA Software.

6.4 MATLAB Example

This section covers the MATLAB examples provided with the M90XA Software. Two console application example programs are provided, one for use with the M9391, and another for use with the M9393. These example programs work with the 32-bit version of MATLAB only. The steps below are for the M9391 but can be applied to the M9393.

To open this example, first copy the example folder files from the installed location to a temporary folder (for eg, C:\Temp). Next, in the *MATLAB* folder, double-click the *ML_UsingM9391.m* file to open it.

6.4.1 Running the MATLAB Example

When you run the *ML_UsingM9391.m* file, you are prompted for the IVI Alias name and whether you would like to run the program in simulation mode.

The IVI Alias name must match the name of a saved connection to the M9391 hardware.

The following image displays an output after running the *ML_UsingM9391.m* example program:

```

Command Window
>> ML_UsingM9391

IVI Alias Name: M9391

Simulate? (T/F)T
Sharable IVI driver session successfully created.
Driver Initialized
Simulate:      True
      1
      * Telnet Port = 5023
      * Socket Visa Address = TCIP0::SMS-DEMO01::5025::SOCKET
      * Socket port = 5025
      * HiSLIP Visa Address = TCIP0::SMS-DEMO01::hislip0::INSTR
      * HiSLIP Device Number = 0
Loading WLAN Application

ErrorQuery: 0, No error.

Enter "i" to use IVI driver or "x" to us M90XA (Enter "q" to quit)i
Using IVI Driver
IVI driver channel power: -56.21 dBm / 10.000 MHz at CF = 2412.000000 MHz
Enter "i" to use IVI driver or "x" to us M90XA (Enter "q" to quit)x
opening up M90XA
mode = <WLAN>
M90XA channel power: -46.32 dBm / 10.000 MHz at CF = 2412.000000 MHz
Enter "i" to use IVI driver or "x" to us M90XA (Enter "q" to quit)q
Quitting
Closing M90XA
Finished.
fx >>

```


If you encounter errors while running the MATLAB examples, you might need to copy the newer .dlls to your MATLAB application folder. Refer to the **Shared Native .dlls** section for more information.

6.4.2 Code Walkthrough

1. Add the .NET assemblies required for M90XA and M9391:

```
%Imports
NET.addAssembly('System');
NET.addAssembly('Agilent.AgM90XA.Interop');
NET.addAssembly('Agilent.AgM9391.Interop');
NET.addAssembly('Ivi.Visa.Interop');
NET.addAssembly('Agilent.M9000');
import Agilent.M9000.Interfaces.*;
import Agilent.M9000.*;
import Agilent.AgM9391.Interop.*
```

2. Open a session to the M9000 Resource Manager.

```
try
    myM9000Session = SessionFactory.OpenM9000Session();
catch exception
    disp(getReport(exception));
    return
end
```

3. Create a sharable M9391 IVI driver session.

```

%Open the IVI driver as a sharable resource
ResourcePluginType = 'AgM9391.Ivi';
iviAlias = iviAliasName;           % Saved Configuration name
hardware
[result, resourceName] =
myM9000Session.OpenSharableResource(ResourcePluginType, iviAlias,
simulate);
if result ~= ResourceManagerResult.Ok
    disp('Could not open sharable resource of type')
    return
else
    disp('Sharable IVI driver session successfully created.')
end

```

4. Start the M90XA Software.

```

% Create driver instance
try
    myDriver = instrument.driver.AgM90XA();
catch exception
    disp(getReport(exception));
    Cleanup();
    return
end

%Initialize the M90XA driver
resourceDesc = char(resourceName);
idquery = true;
reset = true;

myDriver.Initialize(resourceDesc, idquery, reset, initOptions);
if myDriver.Initialized
    disp('Driver Initialized');
else
    disp('Driver Initialized FAILED');
    return
end

```

5. Switch between the M90XA Software and direct access to the M9391 driver session by sharing the M9391 IVI driver session. The private method `UseIviDriver()` shows how to checkout the hardware driver from Agilent M9000 Resource Manager, use it, then checkin back so that it is available for use.

```
global resourceDesc;
global myM9000Session;

ACQUISITION_BANDWIDTH = 10.0e6;
ACQUISITION_DURATION = 1.0e-3;

% Make the measurement at the current center frequency of the M90XA
cf = QueryM90xaParameter(':FREQ:CENT?');
bw = ACQUISITION_BANDWIDTH;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[result, m9000Resource]= myM9000Session.Checkout(resourceDesc);
M9391driver = (m9000Resource.Item); %Agilent.AgM9391.Interop.IAgM9391
```

```

M9391driver.RestoreDefaultProperties();
M9391driver.AcquisitionMode =
Agilent.AgM9391.Interop.AgM9391AcquisitionModeEnum.AgM9391AcquisitionModePo
wer; % 2 = AgM9391AcquisitionModePower
M9391driver.RF.Power = 0.0; % dBm
M9391driver.IAgM9391RF_Frequency = cf;
M9391driver.PowerAcquisition.Configure( 10.0e6 , 1.0e-3); %(Bandwidth,
seconds duration)
M9391driver.IAgM9391PowerAcquisition_Bandwidth = bw;
M9391driver.PowerAcquisition.Duration = ACQUISITION_DURATION;
M9391driver.Apply();
M9391driver.Arm();

%Getting measurement
power = -200;
overload = false;
[power, overload] = M9391driver.PowerAcquisition.ReadPower(0,
Agilent.AgM9391.Interop.AgM9391PowerUnitsEnum.AgM9391PowerUnitsdBm, power,
overload);
fprintf('IVI driver channel power: %0.2f dBm / %0.3f MHz at CF = %0.6f
MHz', power, bw/1e6, cf/1e6);

```

```
myM9000Session.Checkin(m9000Resource);
```

Internally, the M90XA is also checking out the same resource, using it, and checking it back in.

6.5 C or C++ Example

The *C_UsingM9391* and *C_UsingM9393* example folders contain C/C++ example programs. These program show the procedure to call M9000 and M90XA library functions from unmanaged (or native) Windows programming environment. *C_UsingM9391* is for use with the M9391 hardware. *C_UsingM9393* is for use with the M9393 hardware. The steps below are for the M9391 but can be applied to the M9393.

This example is a console program that performs the following steps:

1. Open an M9000 session. This will be used to manage the sharing of VSA hardware and driver resources.
2. Use the M9391 IVI-C driver to open a VSA driver session.
3. Add the VSA driver IVI session to the M9000 resource pool.
4. Use the M90XA IVI-C driver to open a M90XA application.
5. Lock the VSA driver resource for use by this program.
6. Use the driver C functions to measure channel power.

7. Unlock the VSA driver resource.
8. Use the M90XA application to measure modulation accuracy.
9. Close driver sessions.

6.5.1 Building the C/C++ Example

To open this example, first copy the example folder files from the installed location to a temporary folder (for eg, C:\Temp).

1. Open the *C_UsingM9391* project file with Microsoft Visual Studio.

This example supports Microsoft Visual C++ 2010, so you might be prompted to update/upgrade the project.

2. Open the source code *C_UsingM9391.c* and review the code.
3. Build the *C_UsingM9391* project.

Compiling requires several .h include files that are installed by the M90XA software, M9391 driver, and VISA (from Agilent IO Libraries). These are found through environment variables such as *IVIROOTDIR32* and *IXIPNPPATH* which are set by the installers. Linking requires these same environment variables to find library files.

6.5.2 Running the C/C++ Example

You can either run the program from within Visual Studio, or find the *.exe* and execute it.

The following output appears:

```

C:\tmp\Examples\C_UsingM9391\Win32\Debug\C_UsingM9391.exe
C_UsingM9391
Example of using the Keysight M9000 API with the M9391 USA.
Runs in simulated mode if no command line parameter is provided.
Run with resource string argument to use hardware.

Initializing M9000 Modular Applications library...
Initializing M9391 USA driver using resource name 'MyUSA'...
USA driver initialized.
USA driver session added to the M9000 resource pool.
Starting M90XA application using the same USA driver...
Opening SCPI connection to M90XA at address 'TCPIP0::5CB31200Q4::hislip0::INSTR'

Locking the USA driver resource for IUI-C use.
Measuring channel power by calling USA IUI-C driver functions.
Measured power -38.43 dBm at 1000.00 MHz
Unlocking the USA driver resource to allow M90XA to use it.
Measuring modulation accuracy with M90XA...
Measured EVM: 999 average, 999 peak

Done - Press Enter to Exit_

```

In addition to the above, the M90XA Software UI appears.

In case you have not specified any command line parameters, the program will use simulated hardware.

To run the program with M9391A hardware:

1. Use the M9391 SFP to select PXI modules.
2. Save the configuration with a name such as, **MyVSA** or **M9391**.

Provide this name as a command line parameter.

6.5.3 Deploying a C/C++ Program

Although this is a native C/C++ program, running the executable requires M90XA and M9391 drivers to be installed, which require version 4 of the .NET Common Language Runtime. The app.config file (as displayed below) has a startup configuration to allow interoperability with older components that were developed for Version 2 of the CLR.

If you are starting from scratch and an app.config file does not exist in your solution, right click on the startup project name and select add new item. From the list of Visual C# items, select “Application Configuration File.” Copy the following text into the App.config file:

```

<?xml version="1.0"?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
</configuration>

```

```

</startup>
</configuration>

```

If you are modifying an existing solution you may need to add the startup setting to the configuration section of the existing solution's app.config file.

6.6 C (Sharp) Example

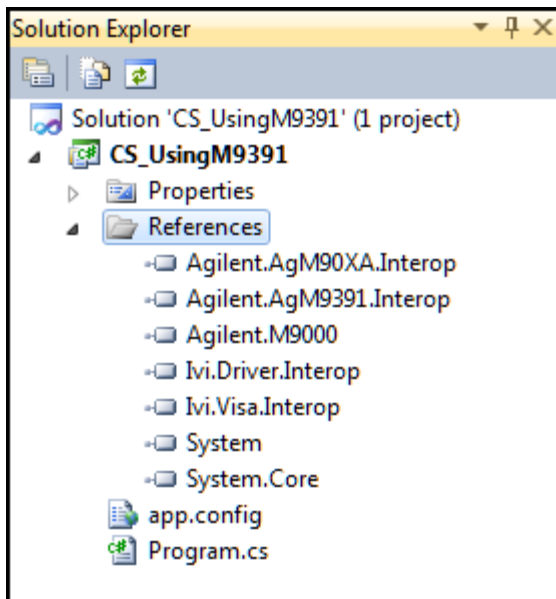
This section covers the *CS_Using* examples provided with the M90XA Software. Two console application example programs are provided, one for use with the M9391, and another for use with the M9393. The steps below are for the M9391 but can be applied to the M9393.

Copy the example folder to another location that you will use for development. This will ensure that you always have access to the original examples.

In the *CS_UsingM9391* folder, double-click the *CS_UsingM9391* project file to open it in Visual Studio.

References

This example references the assemblies as displayed below:

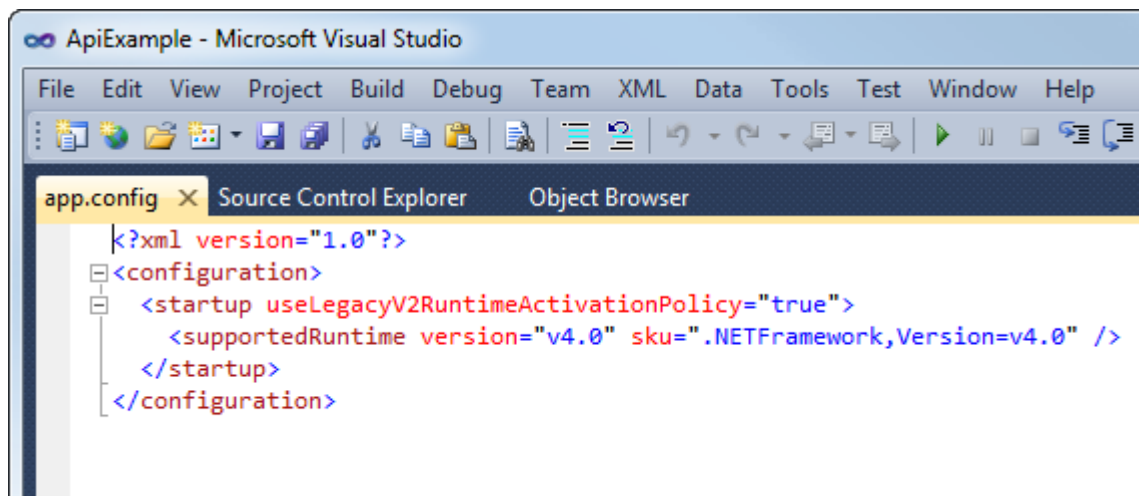


Assembly	Purpose	Copy Local
Agilent.AgM90XA.Interop	The Agilent M90XA IVI-COM driver	False
Agilent.AgM9391.Interop	The Agilent M9391 IVI-COM driver	False

Assembly	Purpose	Copy Local
Agilent.M9000	The Agilent M9000 Application Framework	False
Ivi.Driver.Interop	For Ivi driver communications	False
Ivi.Visa.Interop	For SCPI communications	False

App.Config

The Agilent M9000 Resource Manager uses version 4.0 of the .NET Common Language Runtime. The app.config file (as displayed below) has a startup configuration to allow interoperability with older components that were developed for Version 2 of the CLR.



If you are starting from scratch and an app.config file does not exist in your solution, right click on the startup project name and select add new item. From the list of Visual C# items, select "Application Configuration File." Copy the following text into the App.config file:

```
<?xml version="1.0"?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
</configuration>
```

If you are modifying an existing solution you may need to add the startup setting to the configuration section of the existing solution's app.config file.

6.6.1 Code Walkthrough

1. Open a session to the M9000 Resource Manager.

```
try
{
    _myM9000Session = SessionFactory.OpenM9000Session();
}
catch (Exception e)
{
    Console.WriteLine("Could not start M9000Session (details follow).");
    Console.WriteLine(e.ToString());
    WaitForUser();
    Cleanup();
    return 1;
}
Console.WriteLine("M9000Core sucessfully started.");
```

2. Create a sharable M9391 IVI driver session.

```
// Open the IVI driver as a sharable resource
var result = _myM9000Session.OpenSharableResource(ResourcePluginType,
_iviAlias, out _resourceName, simulate);
if (result != ResourceManagerResult.Ok)
{
    Console.WriteLine("Could not open sharable resource of type '" +
ResourcePluginType
    + "' using alias '" + _iviAlias + "'. \n" + result.ToString() + ".
See Windows Event Log.");
    WaitForUser();
    Cleanup();
    return 2;
}
```

3. Start the M90XA Software.

```
try
{
    Console.WriteLine("Starting the M90XA, please be patient.");
    // Start the M90XA
    _myM90XA = new AgM90XA();
}
catch (Exception e)
{
    Console.WriteLine("Could not instantiate the M90XA (details follow).");
    Console.WriteLine(e.ToString());
    WaitForUser();
    Cleanup();
    return 3;
}

string iviOptions = "Simulate=true";
string optionString = iviOptions;
try
{
    _myM90XA.Initialize(_resourceName, false, false, optionString);
}
catch (Exception e)
{
    Console.WriteLine("Could not initialize the M90XA (details follow).");
    Console.WriteLine(e.ToString());
    WaitForUser();
    Cleanup();
    return 3;
}
```

4. Switch between the M90XA Software and direct access to the M9391 driver session by sharing the M9391 IVI driver session.

The private method `UseIviDriver()` shows how to checkout the hardware driver from Agilent M9000 Resource Manager, use it, then checkin back so that it is available for use.

```
#region Check-out (lock) IVI HW driver for use

M9000Resource m9000Resource;
if ( _myM9000Session.Checkout( _resourceName, out m9000Resource) !=
ResourceManagerResult.Ok)
{
    Console.WriteLine("Could not checkout driver resource '{0}'",
_resourceName );
    WaitForUser();
    return 1;
}

// Cast it to the type you asked for
var driver = (IAgM9391) m9000Resource.Item;

Console.WriteLine("\nDriver is available for use.");

#endregion
```

```
#region Use IVI HW driver resource

// Note that if we are doing multiple checkout/checkin we may be able to
skip
// some setup stuff if we know we were the last user.
if (!m9000Resource.StateQuestionable)
{
    Console.WriteLine("No one used the driver since we last unlocked it.");
}

driver.RestoreDefaultProperties();
driver.AcquisitionMode =
AgM9391AcquisitionModeEnum.AgM9391AcquisitionModePower;
driver.PowerAcquisition.Configure( 10.0e6 /* MHz BW */, 1.0e-3 /* seconds
duration */ );
driver.Apply();
driver.Arm();
double power = -200.0;
bool overload = false;
driver.PowerAcquisition.ReadPower( 0,
AgM9391PowerUnitsEnum.AgM9391PowerUnitsdBm, ref power, ref overload );
Console.WriteLine("IVI driver power measurement result: {0:G4} dBm",
power);

#endregion
```

```
#region Check-in (unlock) IVI HW driver so others may use it
_myM9000Session.Checkin(m9000Resource);
Console.WriteLine("\nDriver checked in, ready for use by others");

#endregion
```

Internally, the M90XA is also checking out the same resource, using it, and checking it back in.

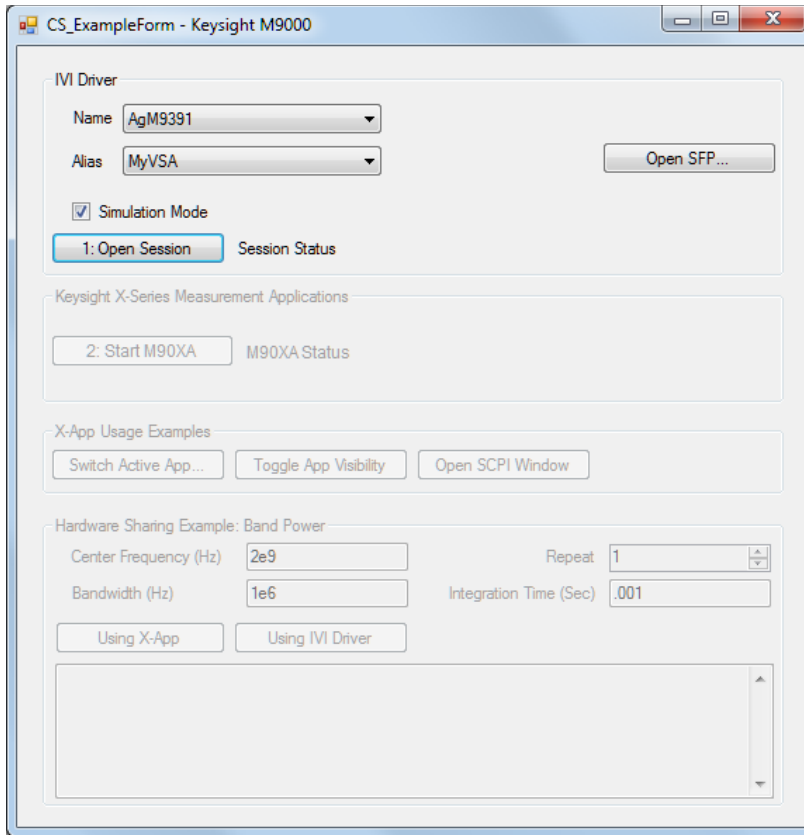
6.7 C (Sharp) Example Form

This section covers the *CS_ExampleForm* programming example provided with the M90XA Software. This example shows how to create a Windows Forms application to start the M90XA Software, select a specific X-series application, perform band power measurements with the M90XA, and perform band power measurements with the IVI driver.

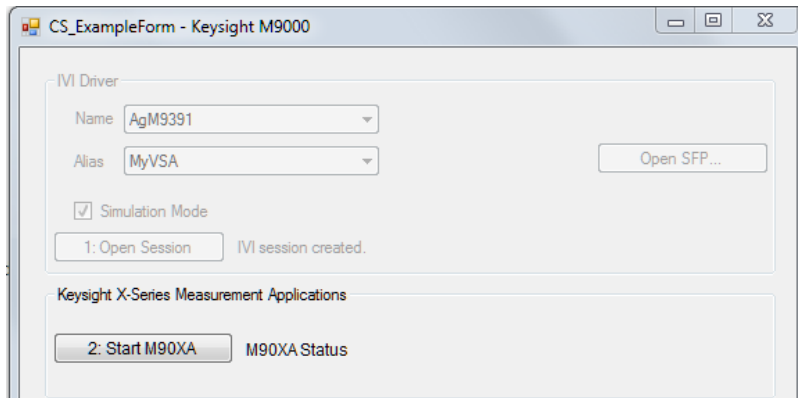
The code for this example is similar to the *CS_UsingM9391* or *CS_UsingM9393* console program.

In addition to source code, a compiled exe is installed. To run it, click **Start > All Programs > Keysight > M90XA X-Series Applications 2.1 > Examples > CS_ExampleForm (64-bit)**.

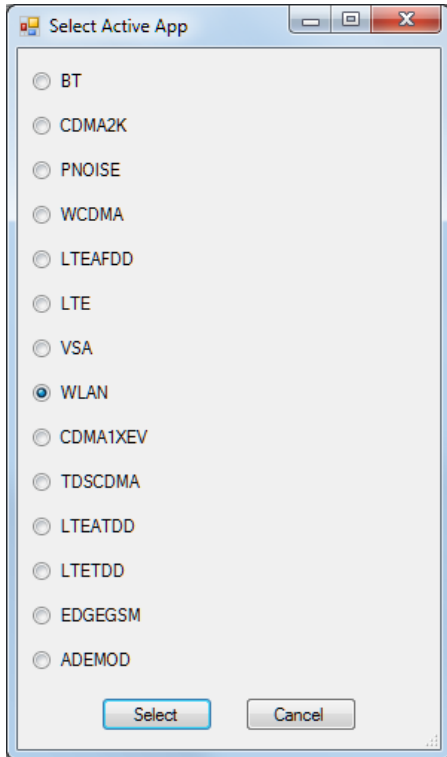
The program first prompts to open the M9391 or M9393 driver session based on an IVI driver **Name**, and a saved hardware configuration **Alias**. Click **Open Session** to open the driver session.



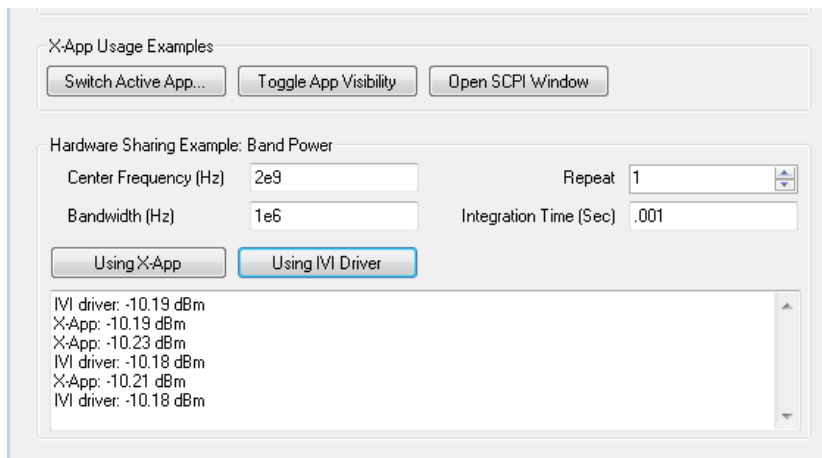
After the sharable IVI resource session is open you may use the driver directly as illustrated by the band power measurements. You may also start the M90XA software by clicking the **Start M90XA** button.



After the M90XA Software is launched, you can use the **Switch Active App...** to switch between the different X-Series Measurement Applications.



In the “Hardware Sharing Example: Band Power” area, click the **Using X-App** button to send commands to the M90XA to perform a band power measurement. This is not supported in some X-series applications so you may need to first switch to WLAN. Click the **Using IVI Driver** button uses IVI driver to perform a band power measurement.



6.7.1 Code Walkthrough

To review or modify this program, make a copy of the *VS.Net* folder, then open the project file for this example in Visual Studio.

The project has the same assembly references and app.config requirements as the *C (Sharp) Example* console programs.

Use Visual Studio to View Code in *MainForm.cs* in the *CS_ExampleForm* project. Near the beginning of this file are some `#define` directives for each supported IVI driver. Unless you have all the IVI drivers installed, you will need to delete or comment-out directives for IVI drivers you are not using.

```
#define M9391_SUPPORTED // remove this line to compile without the M9391 IVI
driver.
#define M9393_SUPPORTED // remove this line to compile without the M9393 IVI
driver.

#if M9391_SUPPORTED
    // This 'using' requires installation of the M9391 IVI driver.  Disable if not
    needed.
    using Agilent.AgM9391.Interop;
#endif

#if M9393_SUPPORTED
    // This 'using' requires installation of the M9393 IVI driver.  Disable if not
    needed.
    using Agilent.AgM9393.Interop;
#endif

using System;
using System.Diagnostics;
using System.Globalization;
using System.Windows.Forms;
using System.Collections.Generic;
using Agilent.M9000;
using Agilent.M9000.Interfaces;
using Ivi.ConfigServer.Interop;
using Ivi.Visa.Interop;
using Microsoft.Win32; // need Agilent.AgM90XA.Interop reference
using Agilent.AgM90XA.Interop;
```

Other portions of this example are similar to the *C (Sharp) Example* console application.

6.8 Visual Basic .Net Example

This section covers the Visual Basic .Net examples provided with the M90XA Software. Two console application example programs are provided, one for use with the M9391, and another for use with the M9393. The steps below are for the M9391 but can be applied to the M9393.

To open this example, first copy the example folder files from the installed location to a temporary folder (for eg, C:\Temp). This will ensure that you always have access to the original examples.

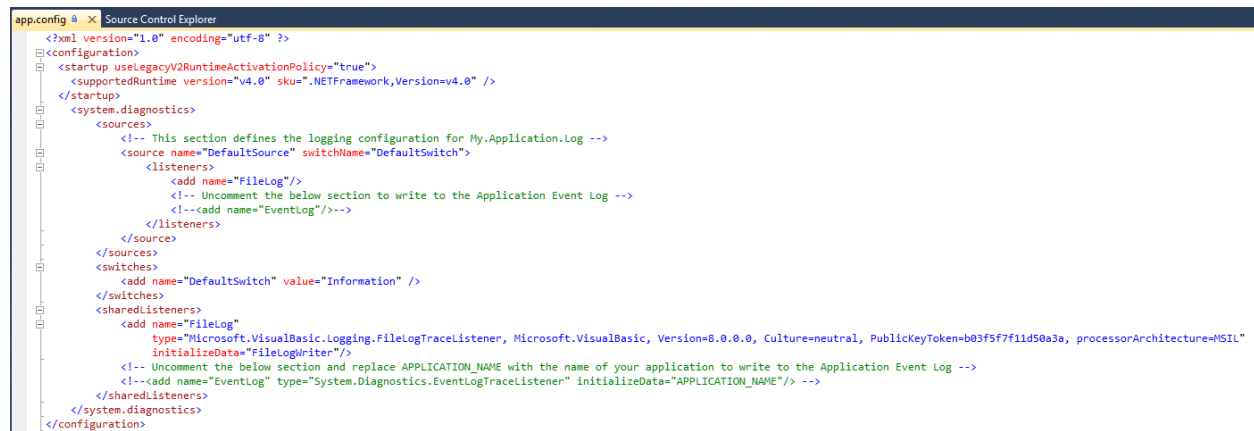
References

This example references the following assemblies:

Assembly	Purpose	Copy Local
Agilent.AgM90XA.Interop	The Agilent M90XA IVI-COM driver	False
Agilent.AgM938x.Interop	The Agilent M938x IVI-Com driver	False
Agilent.AgM9391.Interop	The Agilent M9391 IVI-COM driver	False
Agilent.M9000	The Agilent M9000 Application Framework	False
Ivi.Driver.Interop	For IVI driver communications	False
Ivi.Visa.Interop	For SCPI communications	False

App.Config

The Agilent M9000 Resource Manager uses version 4.0 of the .NET Common Language Runtime. The app.config file (as displayed below) has a startup configuration to allow interoperability with older components that were developed for Version 2 of the CLR.



```

app.config  Source Control Explorer
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
  <system.diagnostics>
    <sources>
      <!-- This section defines the logging configuration for My.Application.Log -->
      <source name="DefaultSource" switchName="DefaultSwitch">
        <listeners>
          <add name="FileLog"/>
          <!-- Uncomment the below section to write to the Application Event Log -->
          <!--<add name="EventLog"/>-->
        </listeners>
      </source>
    </sources>
    <switches>
      <add name="DefaultSwitch" value="Information" />
    </switches>
    <sharedListeners>
      <add name="FileLog"
        type="Microsoft.VisualBasic.Logging.FileLogTraceListener, Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL"
        initializeData="FileLogWriter"/>
      <!-- Uncomment the below section and replace APPLICATION_NAME with the name of your application to write to the Application Event Log -->
      <!--<add name="EventLog" type="System.Diagnostics.EventLogTraceListener" initializeData="APPLICATION_NAME"/>-->
    </sharedListeners>
  </system.diagnostics>
</configuration>

```

If you are starting from scratch and an app.config file does not exist in your solution, create one and copy the following text into the App.config file:

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
  <system.diagnostics>
    <sources>
      <!-- This section defines the logging configuration for
My.Application.Log -->
      <source name="DefaultSource" switchName="DefaultSwitch">
        <listeners>
          <add name="FileLog"/>
          <!-- Uncomment the below section to write to the Application

```



```

Event Log -->
    <!--<add name="EventLog"/>-->
    </listeners>
  </source>
</sources>
<switches>
  <add name="DefaultSwitch" value="Information" />
</switches>
<sharedListeners>
  <add name="FileLog"
    type="Microsoft.VisualBasic.Logging.FileLogTraceListener,
Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL"
    initializeData="FileLogWriter"/>
  <!-- Uncomment the below section and replace APPLICATION_NAME with the
name of your application to write to the Application Event Log -->
  <!--<add name="EventLog"
type="System.Diagnostics.EventLogTraceListener"
initializeData="APPLICATION_NAME"/> -->
</sharedListeners>
</system.diagnostics>
</configuration>

```

If you are modifying an existing solution you may need to add the startup setting to the configuration section of the existing solution's app.config file.

6.8.1 Building the VB.Net Example

1. Open the *VB_UsingM9391* project file.
2. Open the source code *VB_UsingM9391.vb* and review the code.
3. Build the *VB_UsingM9391* project.

Compiling requires several .h include files that are installed by the M90XA software, M9391 driver, and VISA (from Agilent/Keysight IO Libraries). These are found through environment variables such as *IVIROOTDIR32* and *VXIPNPPATH* which are set by the installers. Linking requires these same environment variables to find library files.

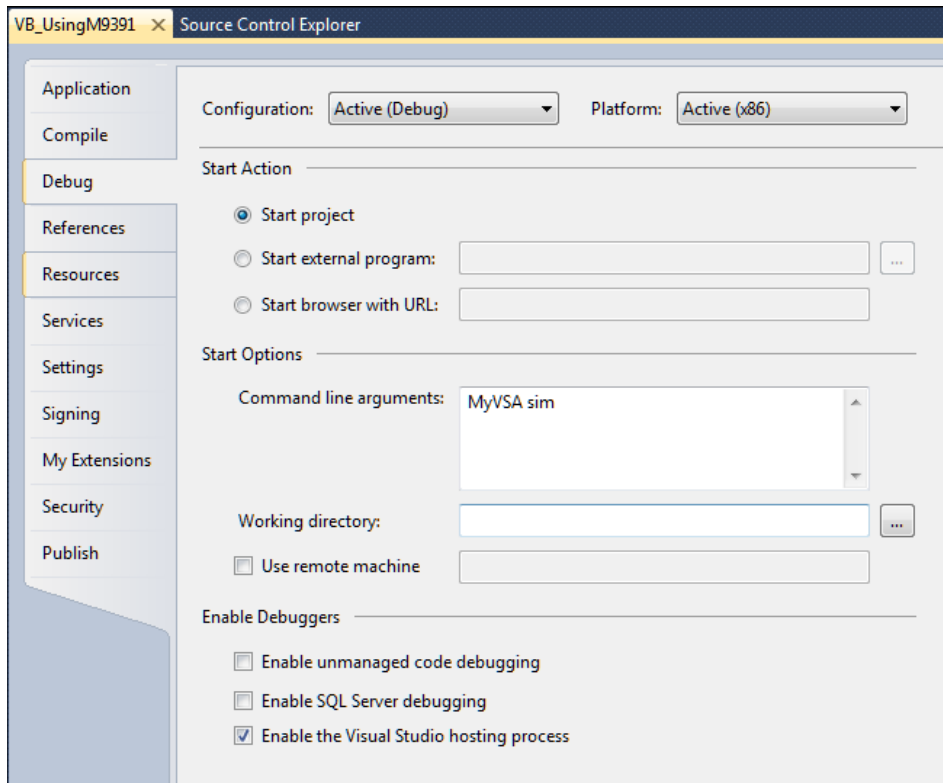
6.8.2 Running the VB.Net Example

You can either run the program from within Visual Studio, or find the .exe and execute it.

You need to specify the following command line parameters:

- The ivi Alias name
- If you want to use/get simulated results add "sim" after the ivi Alias name.

You can also specify these parameters in Visual Studio. Click **Project > Properties**. On the Debug tab, under the Start Options section, specify the command line arguments.



When you run the program, the following output appears:

```
M9000Core successfully started.
Example usage of simulated M9391A IUI driver session using resource 'M9391A'.
Sharable IUI driver session successfully created.
Starting the M90XA, please be patient.
M90XA successfully started.
 * Telnet Port = 5023
 * Socket Visa Address = TCP/IP0::5CB31200Q4::5025::SOCKET
 * Socket port = 5025
 * HiSLIP Visa Address = TCP/IP0::5CB31200Q4::hislip0::INSTR
 * HiSLIP Device Number = 0
Loading WLAN application.

Press 'i' to use IUI driver, or 'x' to use M90XA.
Press the Escape (Esc) key to quit:
iDriver is available for use.
Driver information:
Identifier: AgM9391
Revision: 1.2.415.1
Vendor: Agilent Technologies
Model: M9391A
FirmwareRev: Siml.2.415.1
Simulate: True
IUI driver power measurement result: -35.26 dBm
Driver checked in, ready for use by others

Press 'i' to use IUI driver, or 'x' to use M90XA.
Press the Escape (Esc) key to quit:
xM90XA IDN response: Agilent Technologies,M90XA,NA00000000,M.14.50

Press 'i' to use IUI driver, or 'x' to use M90XA.
Press the Escape (Esc) key to quit:
```

6.8.3 Code Walkthrough

1. Open a session to the M9000 Resource Manager.

```
Try
    _myM9000Session = SessionFactory.OpenM9000Session()
Catch e As Exception
    Console.WriteLine("Could not start M9000Session (details
follow).")
    Console.WriteLine(e.ToString())
    WaitForUser()
    Cleanup()
    Return 1
End Try
Console.WriteLine("M9000Core successfully started.")
```

2. Create a sharable M9391 IVI driver session.

```
' Open the IVI driver as a sharable resource
Dim result =
_myM9000Session.OpenSharableResource(ResourcePluginType, _iviAlias,
_resourceName, simulate)
If result <> ResourceManagerResult.Ok Then
    Console.WriteLine("Could not open sharable resource of type '"
+ ResourcePluginType + "' using alias '" + _iviAlias + "'. " +
result.ToString() + ". See Windows Event Log.")
    WaitForUser()
    Cleanup()
    Return 2
End If
Console.WriteLine("Sharable IVI driver session successfully
created.")
```

3. Start the M90XA Software.

```
Try
    Console.WriteLine("Starting the M90XA, please be patient.")
    ' Start the M90XA
    _myM90XA = New AgM90XA()
Catch e As Exception
    Console.WriteLine("Could not instantiate the M90XA (details
follow).")
    Console.WriteLine(e.ToString())
    WaitForUser()
    Cleanup()
    Return 3
End Try

    Dim iviOptions As String = "Simulate=true"
    Dim optionString As String = iviOptions

Try
    _myM90XA.Initialize(_resourceName, False, False, optionString)
Catch e As Exception
    Console.WriteLine("Could not initialize the M90XA (details
follow).")
    Console.WriteLine(e.ToString())
    WaitForUser()
    Cleanup()
    Return 3
End Try
```

4. Switch between the M90XA Software and direct access to the M9391 driver session by sharing the M9391 IVI driver session.

The private method `UseIviDriver()` shows how to checkout the hardware driver from Agilent M9000 Resource Manager, use it, then checkin back so that it is available for use.

```

' Check-out (lock) IVI HW driver for my use
  Dim m9000Resource As M9000Resource
  If _myM9000Session.Checkout(_resourceName, m9000Resource) <>
ResourceManagerResult.Ok Then
    Console.WriteLine("Could not checkout driver resource '{0}'",
_resourceName)
    WaitForUser()
    Return 1
  End If
  ' Cast it to the type you asked for
  Dim driver = DirectCast(m9000Resource.Item, IAgM9391)
  Console.WriteLine("Driver is available for use.")
'endregion

```

```

' Use IVI HW driver resource
  ' Note that if we are doing multiple checkout/checkin we may be
able to skip
  ' some setup stuff if we know we were the last user.
  If Not m9000Resource.StateQuestionable Then
    Console.WriteLine("No one used the driver since we last
unlocked it.")
  End If
  'Get some driver properties, just as an example
  PrintDriverProperties(driver)
  driver.RestoreDefaultProperties()
  driver.AcquisitionMode =
AgM9391AcquisitionModeEnum.AgM9391AcquisitionModePower
  driver.PowerAcquisition.Configure(10000000.0, 0.001) 'MHz
Bandwidth, second duration
  driver.Apply()
  driver.Arm()
  Dim power As Double = -200.0
  Dim overload As Boolean = False
  driver.PowerAcquisition.ReadPower(0,
AgM9391PowerUnitsEnum.AgM9391PowerUnitsdBm, power, overload)
  Console.WriteLine("IVI driver power measurement result: {0:G4}
dBm", power)

```

```

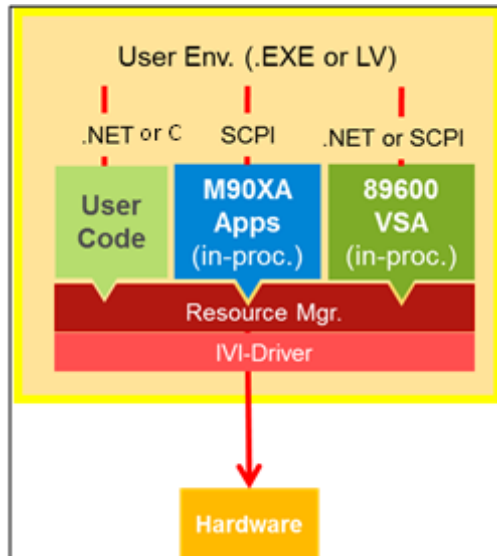
' Check-in (unlock) IVI HW driver so others may use it
_myM9000Session.Checkin(m9000Resource)
Console.WriteLine("Driver checked in, ready for use by others")

```

Internally, the M90XA is also checking out the same resource, using it, and checking it back in.

Chapter 7 – 89600 VSA: Running In-Process and Sharing Hardware Driver Sessions

In some applications it is desirable to use the 89600 VSA Software in-process with the M90XA X Series Applications and/or user programming. The primary advantage is that the hardware driver session may be shared between apps very quickly, nearly eliminating mode switching time.



7.1 Minimum Version Requirements

M90XA X-Series Applications: 2.1

89600 VSA Software: 19.02 (note: if multiple versions of 89600 VSA Software are installed, the newest version will be used)

M9391 PXI Receiver: 1.2.417.1

M9393 PXI Receiver: 1.1.518.1

7.2 Preparing the 89600

Before attempting to run the 89600 in-process with the M90XA, you must prepare the SCPI Connection Type and the Hardware Configuration as described below.

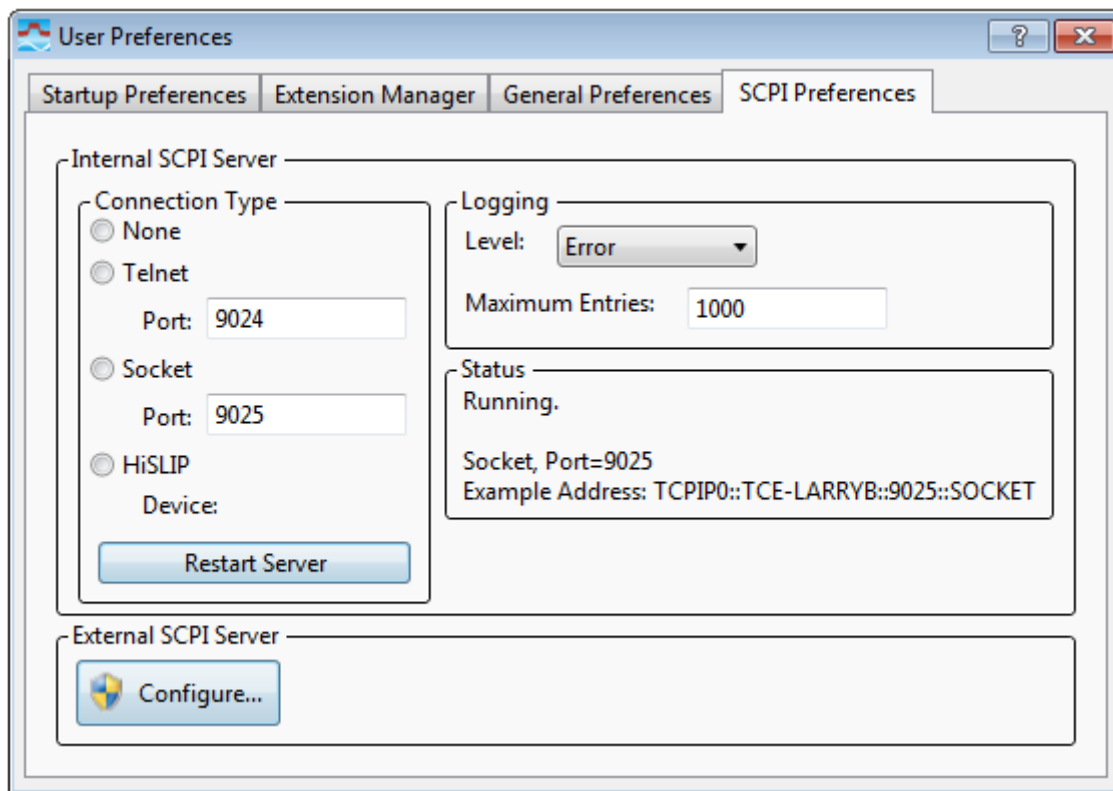
7.2.1 SCPI Preferences

When the 89600 is running in-process with the M90XA and a SCPI connection is being used to communicate with the 89600, conflicts can arise. To avoid potential conflicts:

- The 89600 must *not* be configured to use a HiSLIP connection type.
- If Telnet or Socket connection types are used, the port numbers must *not* conflict with port numbers on the M90XA.
- The M90XA must be started before the 89600 VSA Software.

If the 89600 .NET API is being used then the 89600 connection type should be set to None.

To set the connection type, start the 89600 VSA Software normally, then select **Utilities > SCPI Preferences**. Then, set the Connection Type. The setting will persist when you close the 89600 VSA Software.



7.2.2 Hardware Configuration

The M9391 and M9393 Soft Front Panels save hardware and driver configuration information to the IVI Config Store as named connections. When the M90XA is initialized, this name is used to specify the hardware to use. (For more information, refer to [Launching M90XA Software](#).) This name is also used by the M9000 Resource Manager for sharing the driver instance. (For more information, refer to [C \(Sharp\) Example](#).)

The 89600 VSA Software saves hardware and driver configuration information differently, so it is necessary to create the configuration both ways and then map the IVI logical name to the 89600 configuration name. By convention, these two names must be identical.

When you want to use M9000 Resource Manager for sharing hardware driver session between 89600 VSA and M90XA X-Series Applications and/or user programming, you must pass a command-line argument to 89600 to indicate that Resource Manager sharing should be used along with the configuration to use it with, and the IVI logical name that should be used with it. Following is the syntax for this command-line argument:

```
/M9000:<89600 Configuration Name>=<IVI Logical Name from IVI Config Store>
```

Sharing the M9300A Frequency Reference Module

The Keysight M9300A PXIe Frequency Reference module can be shared across multiple measurement applications. While the M9300A module is being shared, any of the configurations that share this reference can control it fully. However, the user interface of some applications may not reflect M9300A settings made by other applications. For example, the Keysight 89600 software might change it to internal reference, but the M90XA soft keys might not show that.

To work around this situation, perform either of the following two steps:

- Configure the M9300A Reference module the same way in both the applications.
- Leave the M9300A Reference module out of the configuration and use the M9300A Soft Front Panel (SFP) to configure it before starting the apps.

7.3 Starting the 89600

To start the M90XA and 89600 applications from the Start menu, select **All Programs > Keysight > M90XA X-Series Applications 2.1 > Keysight M90XA 2.1 + 89600 VSA**.

To start the M90XA and 89600 applications in-process with your own program, see the programming example [C \(Sharp\) Example Form with 89600](#). There is a shortcut to the example programs on the Start menu at **All Programs > Keysight > M90XA X-Series Applications 2.0 > Examples**.

7.4 Sharing the Hardware

7.4.1 Checkout

To “check out” the shared driver session for use by the 89600 VSA Software, simply set a hardware parameter or start the measurement. The 89600 does the checkout automatically any time the user performs an action that requires hardware access. If the shared driver session is not available because it is checked out to another application, the request times out.

Note that this is different than the way the M90XA X-Series Applications wait patiently for the driver session to become available, while allowing the user to abort the request at any time.

7.4.2 Check in

To “check in” the shared driver session so that it can be used by other applications, select **Control > Disconnect** from the 89600 VSA Software GUI (or `Measurements.SelectedItem.SelectedAnalyzer.Disconnect()`; from the API).

Note that this is different than the way the M90XA X-Series Applications automatically check in the driver session after every hardware access. With the 89600, you must explicitly disconnect from the hardware before the driver session is checked in.

7.5 Programming Example with 89600 VSA

This section covers the available programming example for M90XA and 89600 VSA. For more information about programming with M90XA, refer to [Programming with M90XA Software](#).

- [C \(Sharp\) Example Form with 89600](#)

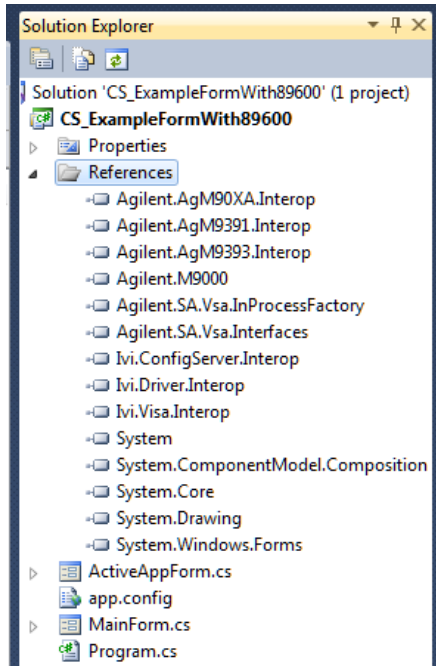
7.5.1 C (Sharp) Example Form with 89600

This section covers the `CS_ExampleFormWith89600` example program provided with the M90XA Software. This example program demonstrates the use of 89600 VSA software in-process with the M90XA X-Series Applications and user programming. For more information, refer to [89600 VSA: Running In-Process and Sharing Hardware Driver Sessions](#).

A single example program caters to both M9391A and M9393A. Create a copy of the *Examples* folder and double-click the `CS_ExampleFormWith89600` project file to open it in Visual Studio.

References

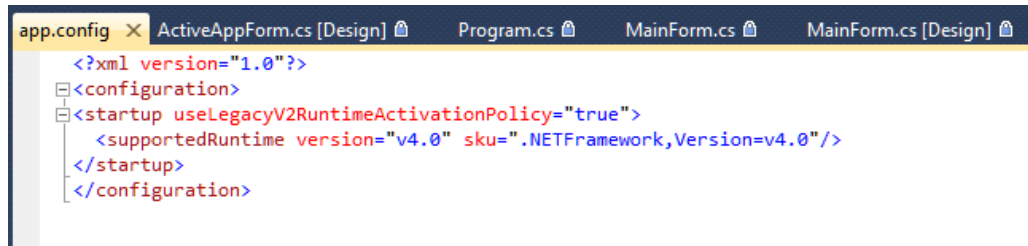
This example references the assemblies as displayed below:



Assembly	Purpose	Copy Local
Agilent.AgM90XA.Interop	The Agilent M90XA IVI-COM driver	False
Agilent.AgM9391.Interop	The Agilent M9391 IVI-COM driver	False
Agilent.AgM9393.Interop	The Agilent M9393 IVI-COM driver	False
Agilent.M9000	The Agilent M9000 Application Framework	False
Agilent.SA.Vsa.InProcessFactory	For starting the most recent installed version of Agilent 89600	True
Agilent.SA.Vsa.Interfaces	The Agilent 89600 interface definitions	False
Ivi.ConfigServer.Interop	For reading the IVI Configuration Store	False
Ivi.Driver.Interop	For IVI driver communications	False
Ivi.Visa.Interop	For SCPI communications	False

App.Config

The Agilent M9000 Resource Manager uses version 4.0 of the .NET Common Language Runtime. The app.config file (as displayed below) has a startup configuration to allow interoperability with older components that were developed for Version 2 of the CLR.

A screenshot of a Visual Studio code editor window. The title bar shows several tabs: 'app.config', 'ActiveAppForm.cs [Design]', 'Program.cs', 'MainForm.cs', and 'MainForm.cs [Design]'. The 'app.config' tab is active, displaying XML code for a configuration file. The code is as follows:

```
<?xml version="1.0"?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
  </startup>
</configuration>
```

If you are starting from scratch and an app.config file does not exist in your solution, right click on the startup project name and select add new item. From the list of Visual C# items, select **Application Configuration File**. Copy the following text into the App.config file:

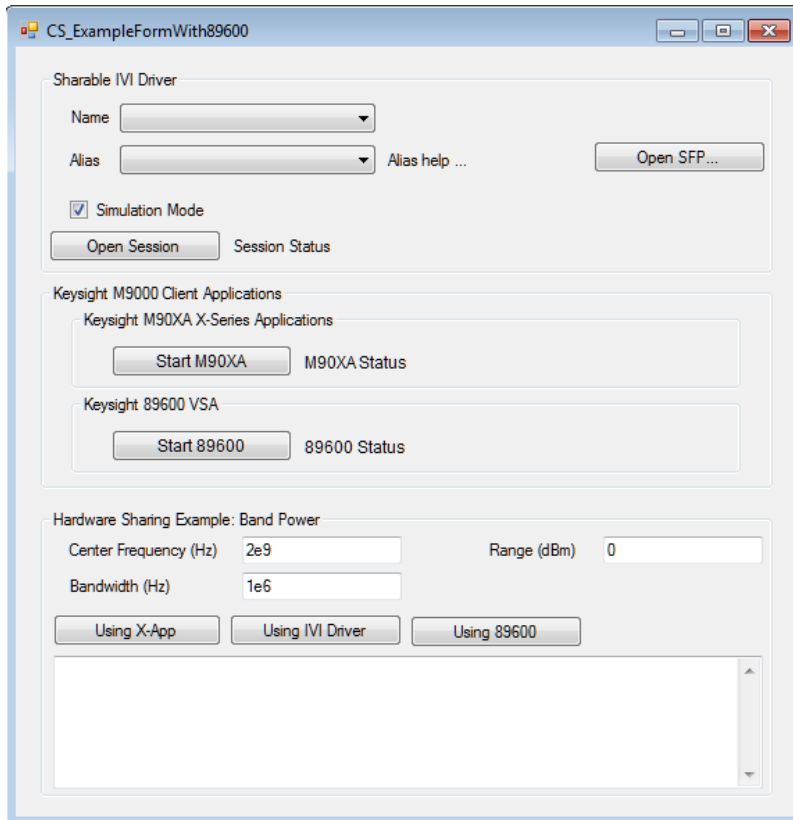
```
<?xml version="1.0"?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
</configuration>
```

If you are modifying an existing solution you may need to add the startup setting to the configuration section of the existing solution's app.config file.

Running the Program

In addition to source code, a compiled exe is installed. To run it, click **Start > All Programs > Keysight > M90XA X-Series Applications 2.1 > Examples > CS_ExampleFormWith89600 (64-bit)**.

The following is a screen capture of the main form.



Here are the steps to run this example program:

1. Open the M9391 or M9393 driver session based on an IVI driver **Name**, and a saved hardware configuration **Alias**. Click **Open Session** to open the driver session.
2. After the sharable IVI resource session is open, you may start the M90XA software by clicking the **Start M90XA** button.
3. Similarly, you may start the 89600 VSA by clicking the **Start 89600** button.
4. Next, specify the values for the parameters for band power measurements.
5. Make measurement using M90XA X-Apps, IVI Driver, or 89600.

To “**check out**” the shared driver session for use by the 89600 VSA Software, simply set a hardware parameter or start the measurement. The 89600 does the checkout automatically any time the user performs an action that requires hardware access. If the shared driver session is not available because it is checked out to another application, the request times out.

Note that this is different than the way the M90XA X-Series Applications wait patiently for the driver session to become available, while allowing the user to abort the request at any time.

To “**check in**” the shared driver session so that it can be used by other applications, select **Control > Disconnect** from the GUI (or `Measurements.SelectedItem.SelectedAnalyzer.Disconnect();` from the API).

Note that this is different than the way the M90XA X-Series Applications automatically check in the driver session after every hardware access. With the 89600, you must explicitly disconnect from the hardware before the driver session is checked in

Code Walkthrough

To review or modify this program, make a copy of the *VS.Net* folder, then open the *CS_ExampleFormWith89600* project file in Visual Studio.

Use Visual Studio to View Code in MainForm.cs in the *CS_ExampleFormWith89600* project. Near the beginning of this file are some #define directives for each supported IVI driver. Unless you have all the IVI drivers installed, you will need to delete or comment-out directives for IVI drivers you are not using.

```
#define M9391_SUPPORTED // remove this line to compile without the M9391 IVI driver.  
#define M9393_SUPPORTED // remove this line to compile without the M9393 IVI driver.
```

This example is similar to the other C# examples in the following areas.

1. Open a session to the M9000 Resource Manager.
2. Create a sharable M9391 IVI driver session.
3. Start the M90XA Software.
4. Switch between the M90XA Software and direct access to the M9391 driver session by sharing the M9391 IVI driver session.

This example adds the Keysight 89600 VSA Software to the process.

1. Start the 89600 VSA Software.
2. Switch between the 89600, M90XA, and direct access to the M9391 driver session by sharing the M9391 IVI driver session.

Starting the 89600 VSA Software

This example is designed so that it does not directly reference the Keysight 89600 VSA Software installed on your computer. This allows you to install a newer version of 89600 at a later date, without having to update a reference and re-compile this program. There are two parts to the 89600 startup, first to create a VSA instance as an object, and second to cast it to an Agilent.SA.Vsa.Application.

```
try  
{  
    // This program references (with Copy Local = false) a fixed version of  
    // Agilent.SA.Vsa.Interfaces.dll which is distributed with the example.  
    // To start the 89600 VSA Software we are calling the InProcessFactory. This  
allows  
    // us to get the latest installed version of the 89600 without referencing the  
    // particular version of Agilent.SA.Vsa.Interfaces.dll directly.
```

```

// This has the advantage that an upgrade to the 89600 VSA Software
// will not break this program.

// Start the latest installed version of the 89600 VSA Software
var success = Setup89600(InProcessFactory.Create(null,-1, 0, commandLineArgs));
if(success)
{
    // M90XA is running; update GUI state
    lbl89600Status.Text = "89600 VSA Running";
    _currentState = _currentState | State.VsaStarted;
    UpdateGui();
}
else
    lbl89600Status.Text = "89600 VSA Did Not Start";
}
catch (Exception ex)
{
    lbl89600Status.Text = ex.Message;
}

```

```

private bool Setup89600(object appObject)
{
    // This is in a private method so that the JIT compiler does not look
    // for VsaApplication before the InProcessFactory has loaded the interfaces
    assembly

    _my89600 = appObject as VsaApplication;
    if (_my89600 != null)
    {
        _my89600.IsExitEnabled = false;
        _my89600.Title += " - Running In-Process with Agilent M90XA";

        // Disconnect the 89600 so it will check in the hardware driver.
        _my89600.Measurements.SelectedItem.SelectedAnalyzer.Disconnect();
        return true;
    }
    return false;
}

```

Switching between 89600, M90XA, and the IVI driver session

Because the M90XA and 89600 VSA applications may be used interactively from the graphical user interface, it is possible that one of these applications has the IVI driver session checked out at the time that the **Using IVI Driver** button is pressed. In the click event handler we force the applications to check in the IVI driver session with the following code:

```

// Stop any running M90XA measurements so it will check in the hardware driver.

```

```
if (_myM90XA != null)
    _myM90XA.Stop();

// Disconnect the 89600 (if necessary) so it will check in the hardware driver.
if (_currentState.HasFlag(State.Vsa))
    Disconnect89600();
```

Likewise, the **Using X-App** event handler disconnects the 89600 and the **Using 89600** event handler stops the M90XA.

Chapter 8 – Troubleshooting

This section covers the procedures to troubleshoot some common problems with M90XA.

- [Common Problems](#)
- [Technical Support](#)

8.1 Common Problems

Here are some common problems you may encounter:

- [M90XA Software fails to start when using 32-bit Applications](#)
- [LTE-Advanced Apps are only available as 64-bit Applications](#)
- [Difficulty starting M90XA](#)
- [Incorrect Alignment Results in M9393](#)

8.1.1 M90XA Software fails to start when using 32-bit Applications

Root Cause

When running M90XA as 32-bit applications, each process can access a maximum of 2 GB memory allocation. This allocation is not sufficient for some situations.

Workaround

To fix this problem set the `LARGEADDRESSAWARE` flag on the application that is expected to use more than 2 GB memory allocation.

This is set by default on the `StartM90XA` binary that is shipped.

However, if you need to build your own binaries, this flag needs to be set. For more details, refer to:

- C/C++: <http://msdn.microsoft.com/en-us/library/wz223b1z%28v=vs.100%29.aspx>
- C#: http://blogs.msdn.com/b/calvin_hsia/archive/2010/09/27/10068359.aspx

8.1.2 LTE-Advanced Apps are only available as 64-bit applications

The LTE-Advanced apps are not supported as 32-bit applications.

8.1.3 Difficulty starting M90XA

If you have difficulty starting the M90XA, you need to first verify that you can connect to the M9391 or M9393 VSA via its Soft Front Panel and that you have a working hardware configuration saved into the IVI

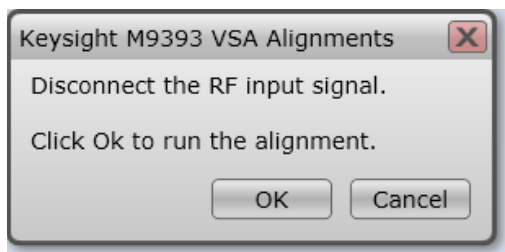
store. To start the M9391 Soft Front Panel, go to **Start>All Programs > Keysight > M9391 > M9391 SFP**. To start the M9393 Soft Front Panel, go to **Start>All Programs > Keysight > M9393 > M9393 SFP**.

For more information about the M9391 SFP, see the M9391 Help available in the same Start menu path (**Start > All Programs > Keysight > M9391 > M9391 Help**).

For more information about the M9393 SFP, see the M9393 Help available in the same Start menu path (**Start > All Programs > Keysight > M9393 > M9393 Help**).

8.1.4 Incorrect Alignment Results in M9393

When using M9393, by default, the measurements are made according to the factory calibration data. If necessary, you can perform additional alignments to improve Amplitude accuracy and LO nulling. Before you perform alignments using M9393, ensure that you disable any RF signal being input to M9393. Any incoming RF signal can alter the alignment results. Hence, when you perform alignments using **Utilities > Alignments**, the following message box is displayed.



You must disconnect the RF input signal and then click **OK** to perform alignments. For more information on Alignments, refer to *Keysight M9393 Soft Front Panel Help*.

8.2 Technical Support

In case of any problems related to the working of M90XA Software contact technical support at www.keysight.com/find/contactus.

myKeysight

myKeysight

www.keysight.com/find/mykeysight

A personalized view into the information most relevant to you.

www.pxisa.org



PCI eXtensions for Instrumentation (PXI) modular instrumentation delivers a rugged, PC-based high-performance measurement and automation system.

Three-Year Warranty

www.keysight.com/find/ThreeYearWarranty

Keysight's commitment to superior product quality and lower total cost of ownership. The only test and measurement company with three-year warranty standard on all instruments, worldwide.



Keysight Assurance Plans

www.keysight.com/find/AssurancePlans

Up to five years of protection and no budgetary surprises to ensure your instruments are operating to specification so you can rely on accurate measurements.



www.keysight.com/go/quality

Keysight Technologies, Inc.

DEKRA Certified ISO 9001:2008

Quality Management System



Keysight Channel Partners

www.keysight.com/find/channelpartners

Get the best of both worlds: Keysight's measurement expertise and product breadth, combined with channel partner convenience.

www.keysight.com/find/M90XA

For more information on Keysight Technologies' products, applications or services, please contact your local Keysight office. The complete list is available at: www.keysight.com/find/contactus

Americas

Canada	(877) 894 4414
Brazil	55 11 3351 7010
Mexico	001 800 254 2440
United States	(800) 829 4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Other AP Countries	(65) 6375 8100

Europe & Middle East

Austria	0800 001122
Belgium	0800 58580
Finland	0800 523252
France	0805 980333
Germany	0800 6270999
Ireland	1800 832700
Israel	1 809 343051
Italy	800 599100
Luxembourg	+32 800 58580
Netherlands	0800 0233200
Russia	8800 5009286
Spain	800 000154
Sweden	0200 882255
Switzerland	0800 805353
	Opt. 1 (DE)
	Opt. 2 (FR)
	Opt. 3 (IT)
United Kingdom	0800 0260637

For other unlisted countries:
www.keysight.com/find/contactus
(BP-09-23-14)



This information is subject to change without notice.
© Keysight Technologies, 2013 - 2015
Published in USA, April 2015
M9000-90001
www.keysight.com