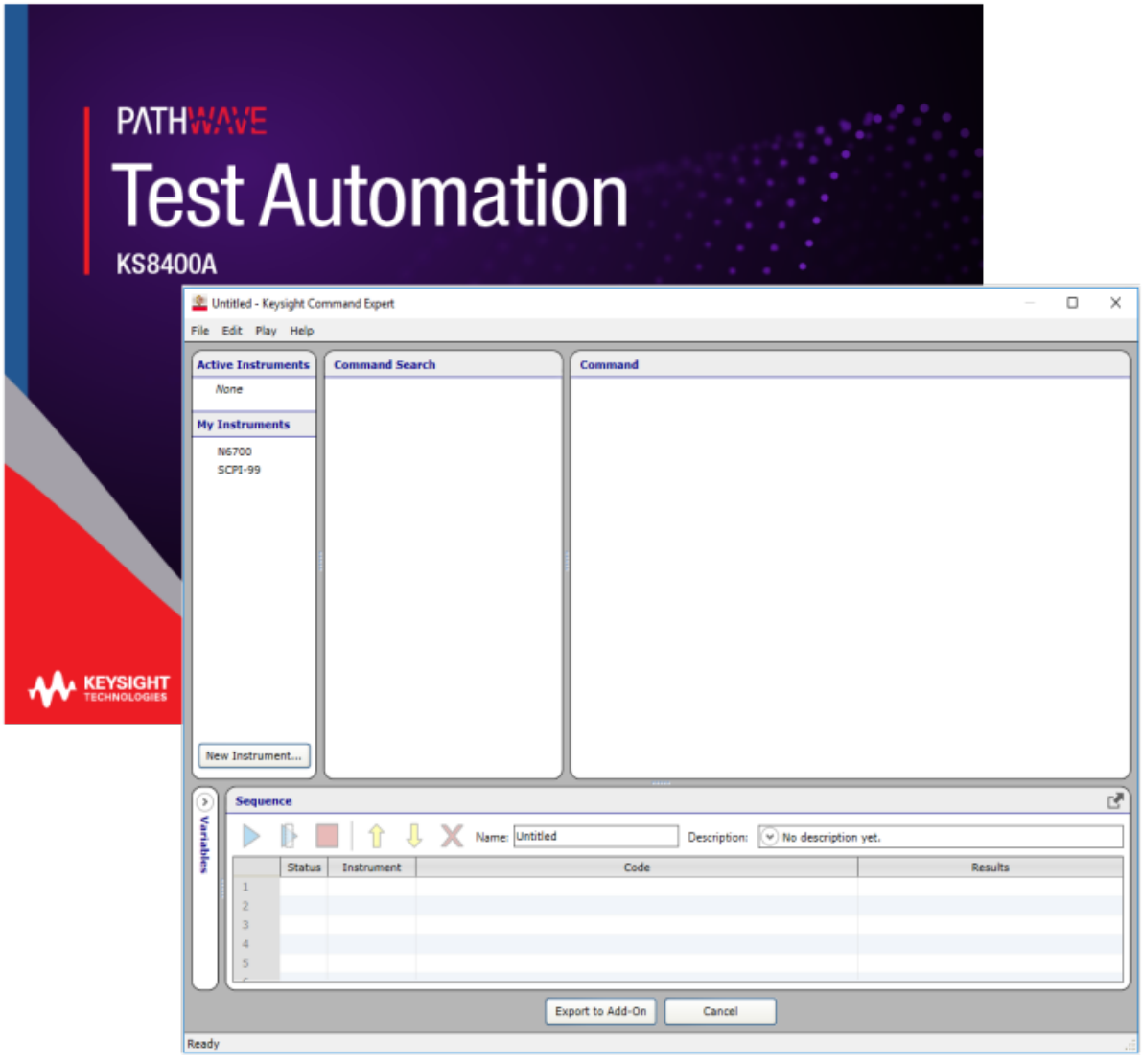
Keysight TAP Training
Lab 3 — Non-Programmer Training



KEYSIGHT TECHNOLOGIES

Notices

DFARS/Restricted Rights Notice

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as “Commercial computer software” as defined in DFAR 252.227-7014 (June 1995), or as a “commercial item” as defined in FAR 2.101(a) or as “Restricted computer software” as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Keysight Technologies’ standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b) (2) (November 1995), as applicable in any technical data.

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 ANY INFORMATION

Technology Licenses

Document History

Published October 31, 2018 for TAP version 8.7

Table of Contents

Overview

TAP is designed to make it easy for programmers to develop test steps and procedures using Visual Studio. If you are a non-programmer, TAP offers some built-in steps that can help you control instruments. This exercise will introduce you to the basic built-in steps and the Command Expert plugin that can be a great productivity enhancer for non-programmers.

Lab Objectives

The objectives of this lab are to:

- Understand how **TAP Test Steps** can be seamlessly created using using **Command Expert** and the **TAP Command Expert Plugin**.
- Learn how the **built-in SCPI steps** can be used for simple instrument control.
- See how to interact with a Keysight analyzer (MXA) and generator (MXG) using TAP. Familiarity with these instruments is NOT a prerequisite.

Prerequisites

This lab requires the following software:

- **Test Automation Platform (TAP)**: Download the latest version from Test Automation Platform Developer's System at www.keysight.com. TAP also requires:
  - **TAP license**, obtained through the TAP GUI upon first start.
  - **Keysight License Manager (KLM)**, installed with TAP and used to manage your license.

  For instructions on installing TAP, see *Keysight TAP Training Lab 1 — TAP GUI Introduction*.

- **Keysight Command Expert** 1.6 or above: Download the installer from Command Expert at www.keysight.com.
- **Keysight IO Libraries** 17.2 or later: Download the installer from [IO Libraries Suite} (https://www.keysight.com/en/pd-1985909/io-libraries-suite?nid=-33330.977662.00&cc=DE&lc=ger) at www.keysight.com.
- **TAP Command Expert Plugin**: Download the plugin from http://plugins.tap.aalborg.keysight.com/. Installing the plugin is explained in this document.

- **MXA Setup.iseq**: a command expert sequence provided to you in an email or on a USB.

Lab Setup

We begin this lab by installing the Command Expert plugin, configuring results listeners, and configuring instruments.

To get started, launch the TAP GUI:

- Select **Windows Start > Keysight Test Automation Platform**, or
- Double-click **TAP_PATH/Keysight.Tap.Gui.exe**.

Note: In this document, **TAP_PATH** refers to the TAP install directory, which is either:

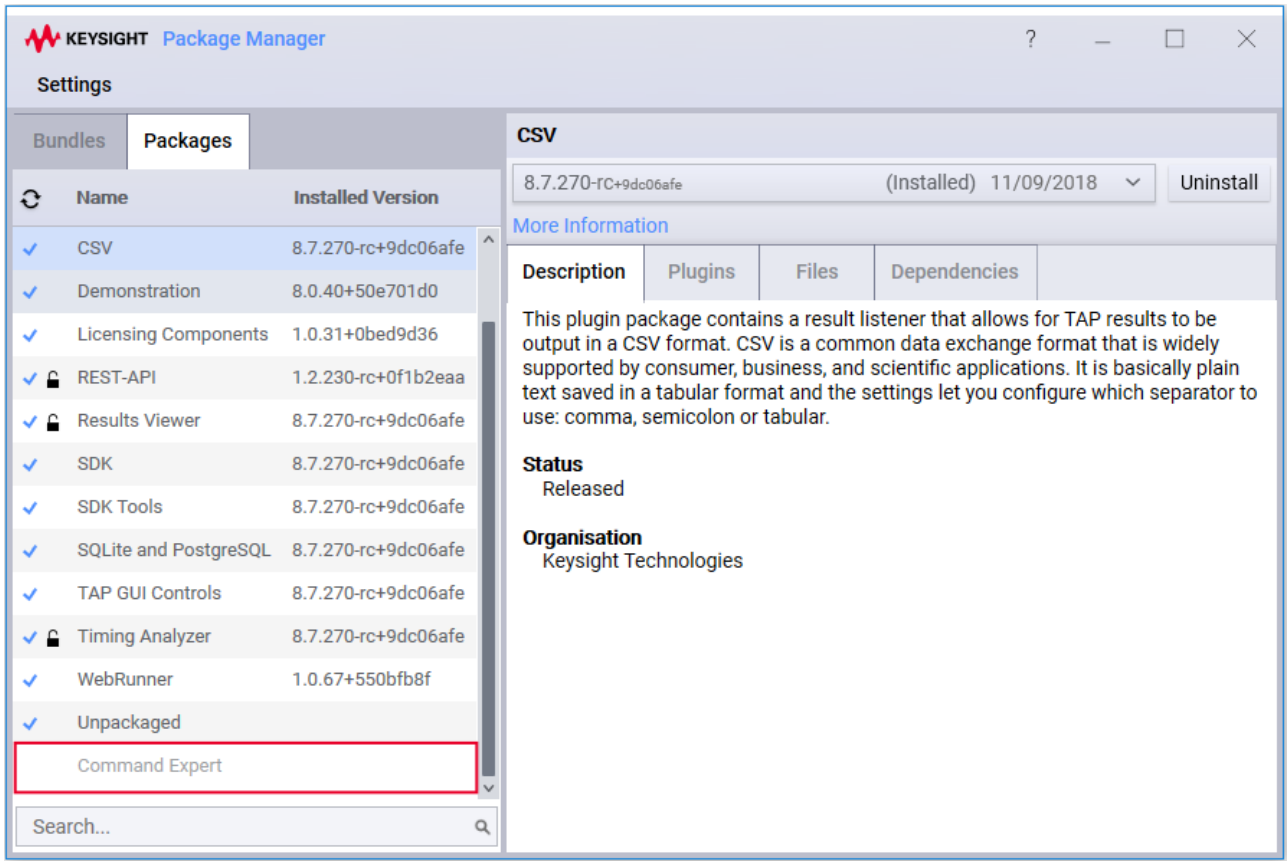
- For **64-bit systems**: C:/Program Files/Keysight/TAP
- For **32-bit systems**: C:/Program Files (x86)/Keysight/TAP

Install TAP Command Expert Plugin

We will install the **TAP Command Expert plugin** using the TAP Package Manger. This plugin makes it possible for non-programmers to leverage Command Expert for creating custom test steps. From Command Expert, you can construct SCPI sequences and define input variables that will show up as properties (Step Settings) on the Command Expert step in the TAP GUI. Results/outputs from Command Expert are automatically stored as a results in TAP, when a database-based Result Listener is configured.

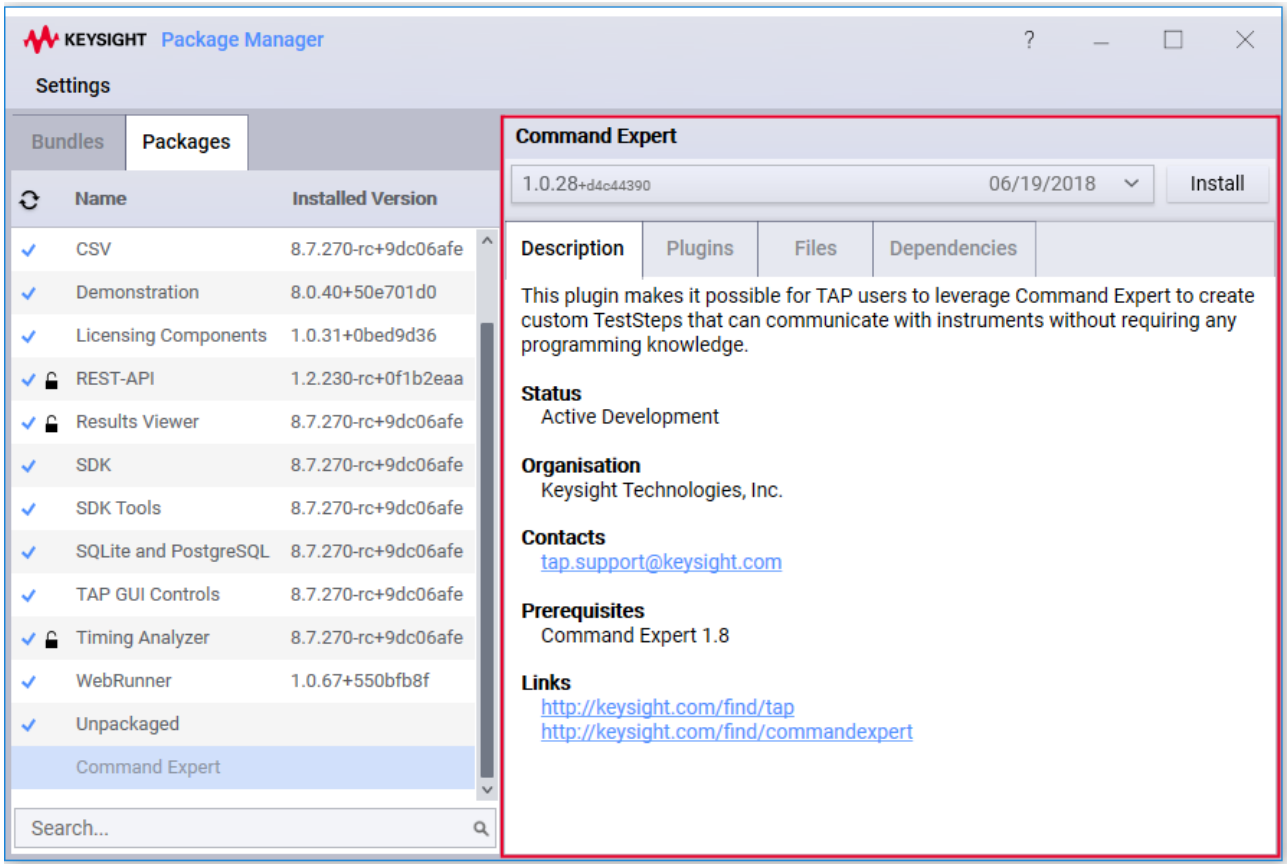
Follow these steps to install the latest version of **TAP Command Expert plugin**:

1. Copy the downloaded Command Expert.*version*.TapPackage to the TAP installation folder (TAP_PATH).
2. In the TAP GUI, select **Tools > Plugin Package Manager** to open **TAP Package Manager**. Notice that the Command Expert plugin appears at the bottom of your list, grayed out, meaning that it is downloaded but not installed:



3. Click on CommandExpert. Now you see the content of the CommandExpert package in the right

side of the window:



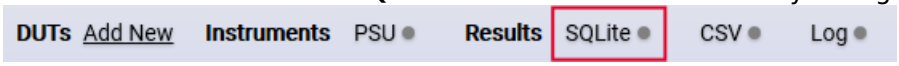
Click the tabs to see the contents:

- The **Description** tab gives a general overview of the plugin.
- The **Plugins** tab show all the custom test steps, Instruments, DUTs, Result Listeners and other components contained in the plugin.
- The **Files** tab shows all the files included in the Package.
- The **Dependencies** tab shows if the plugin is dependent on any other packages.

4. Click **Install**. When the installation completes, close the TAP Package Manager.

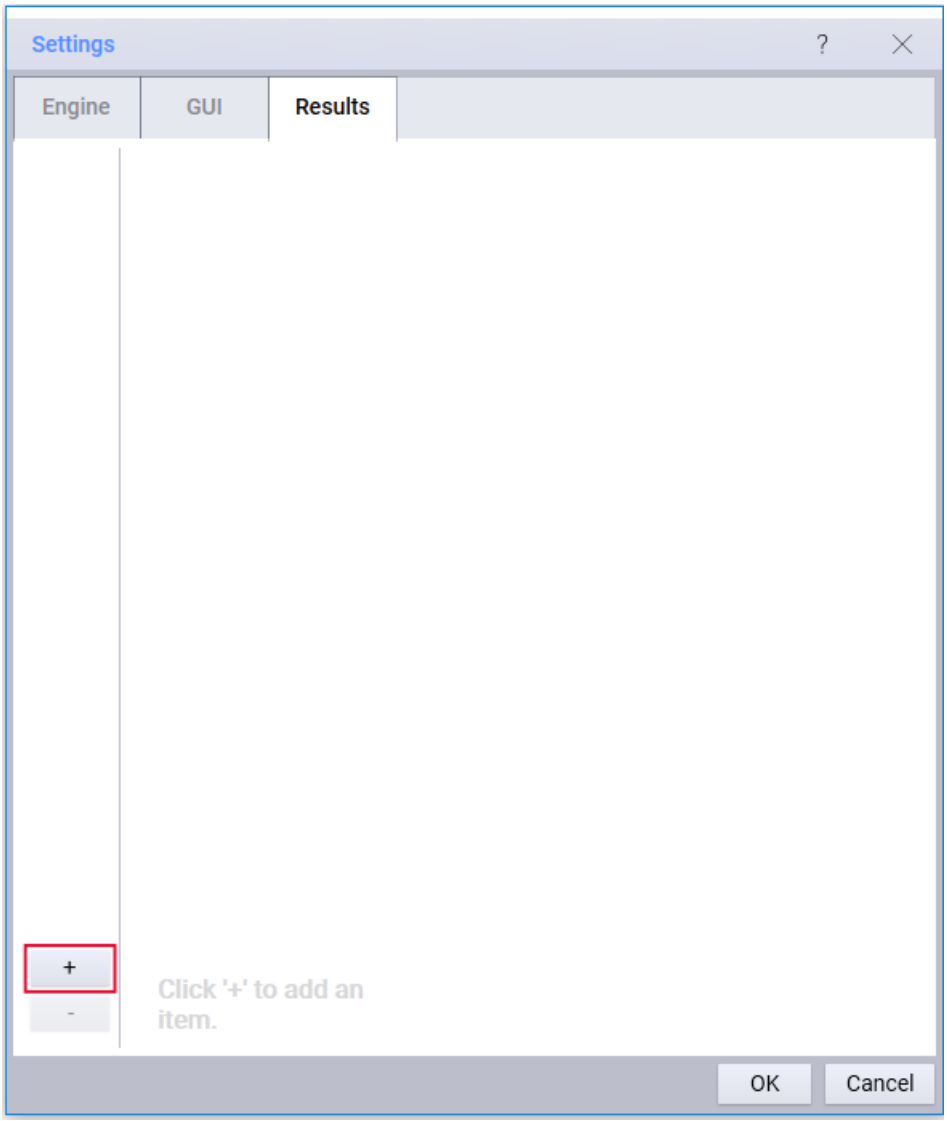
Configure Result Listeners

For this lab you need an **SQLite Result Listener**, which is based on a lightweight database. It consists of a single file (stored on the local system) that encapsulates the full data set. If you already have a **SQLite Result Listener** configured (perhaps from previous lab) you can skip this section. You can see in the **Resource Bar** if an **SQLite Result Listener** is already configured:

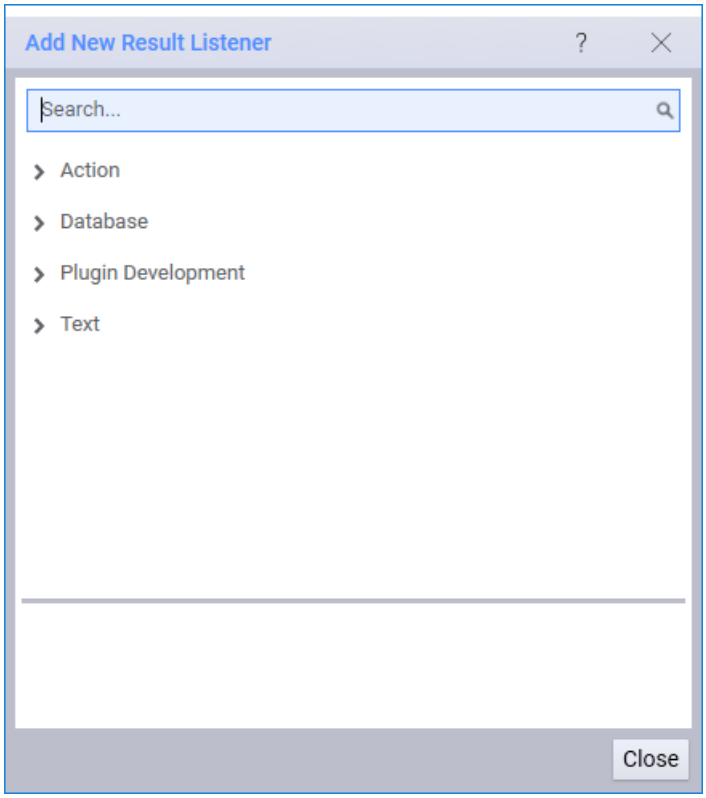


We need to configure where and how TAP stores results. Follow these steps to configure the **TAP Result Listener**.

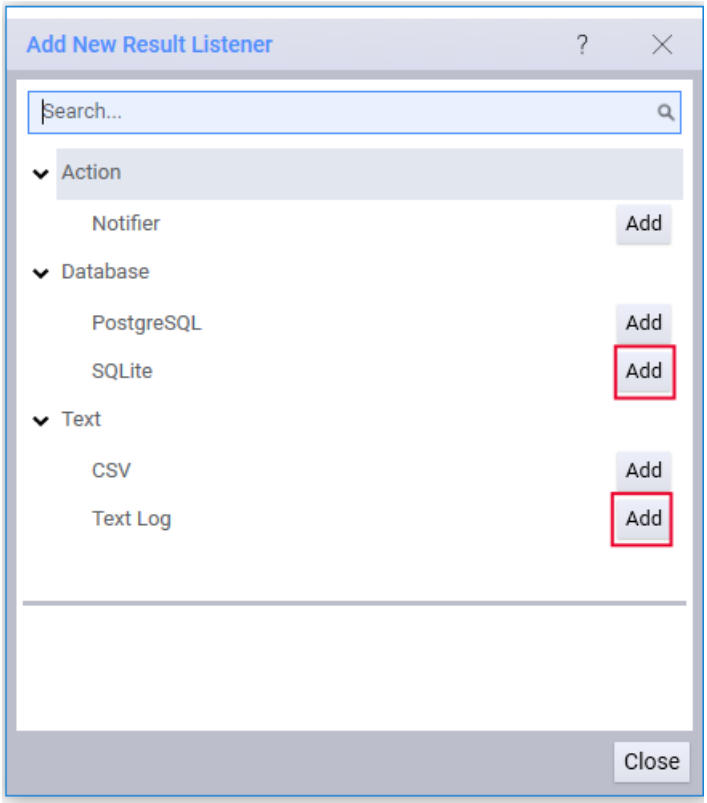
1. In the TAP GUI, select **Settings > Results**. There are no result listeners configured by default.



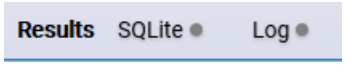
2. Click the **+** in the lower-left corner of the **Settings** window. A window with supported Result Listeners appears.



3. Click the arrows (›) to expand the groups then click **Add** next to **SQLite** to add the **SQLite Result Listener** and click **Add** next to **Text Log** to add a **Text Log Result Listener**. Close the window when done.



Click **OK** in the Settings window. Now you should see two **Result Listeners** in the **Resource Bar** at the bottom of the TAP GUI:



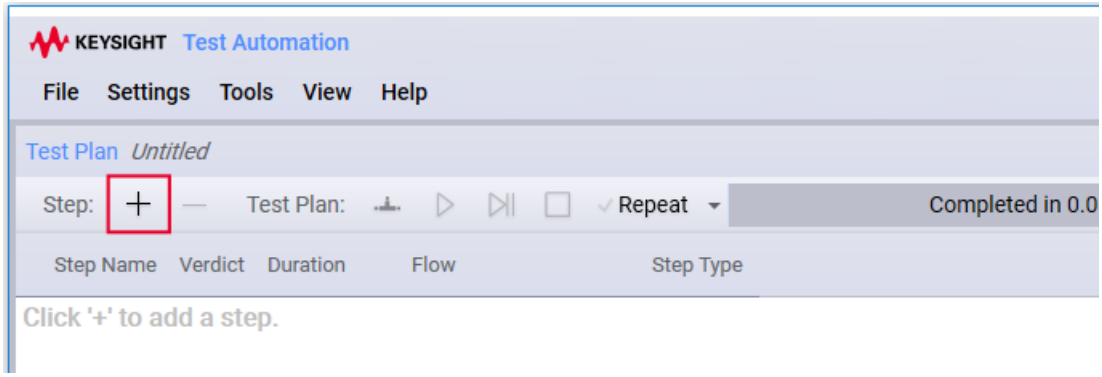
Using Command Expert

Keysight's Command Expert software provides fast and easy instrument control. It combines instrument commands, documentation, syntax checking, command execution, and code generation in one tool. Command Expert works with instruments that use Standard Commands for Programmable Instrumentation (SCPI), IVI-C, or IVI-COM drivers. In this lab we will focus on controlling instruments via SCPI.

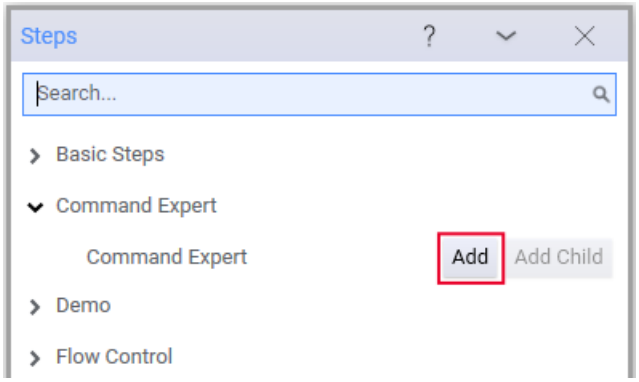
Follow these steps to get familiar with TAP Command Expert plugin.

Accessing Command Expert

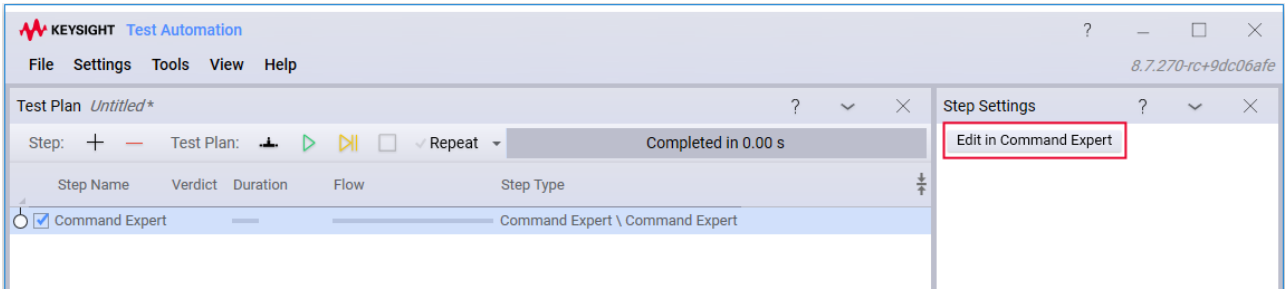
1. In the TAP GUI, select **File > New** to create a new test plan.
2. Add a **Command Expert** step:
    - Click the **+** button to open the **Add New Steps** dialog:



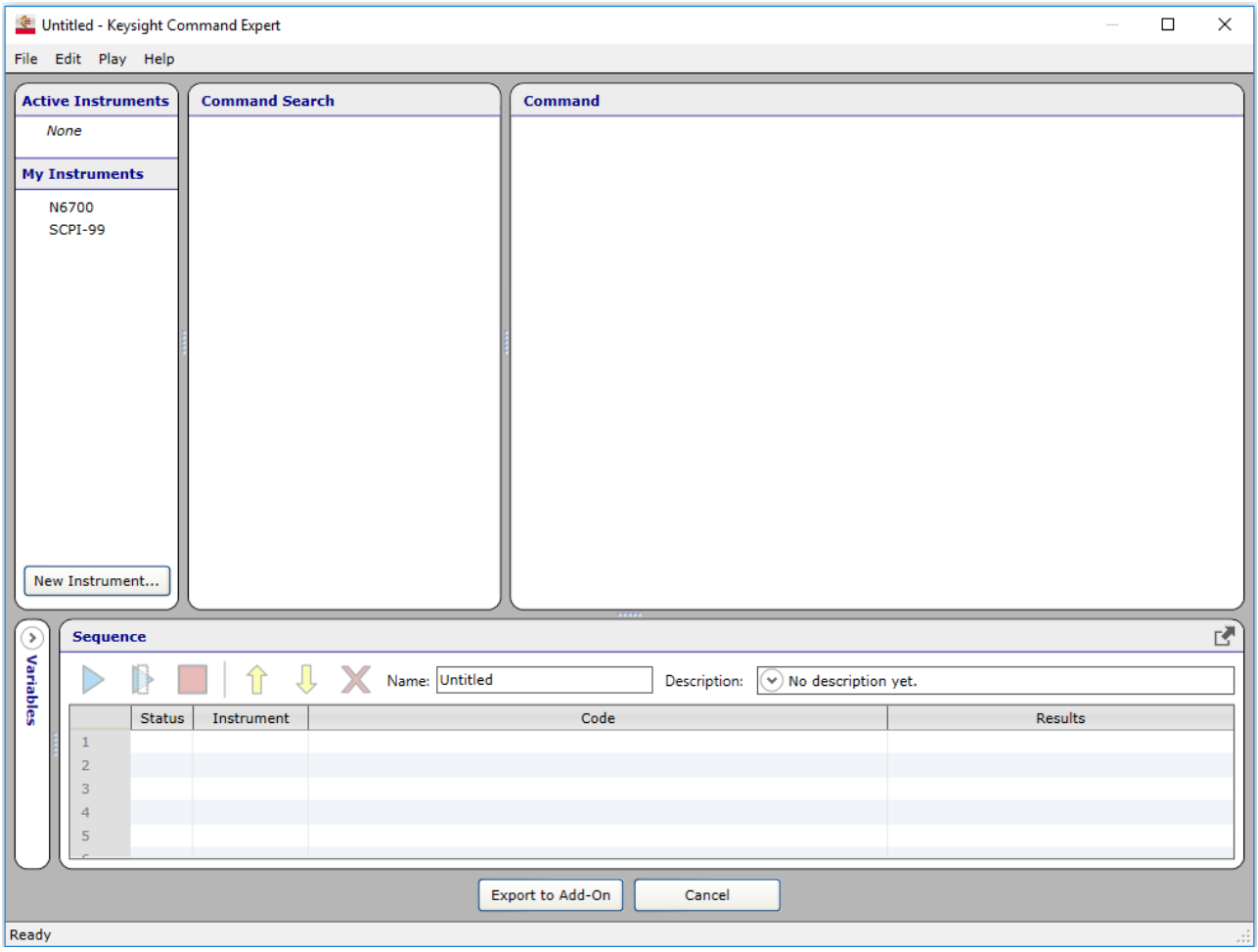
    - Expand the **Command Expert group**, then click **Add** to insert a **Command Expert** step.



    - Close the dialog.
3. You now have a test plan with one **Command Expert** step. Select the step to see the **Step Settings** for this step. Notice that the settings must be edited in **Command Expert**:



4. Click on **Edit in Command Expert** to launch **Command Expert** in add-on mode:



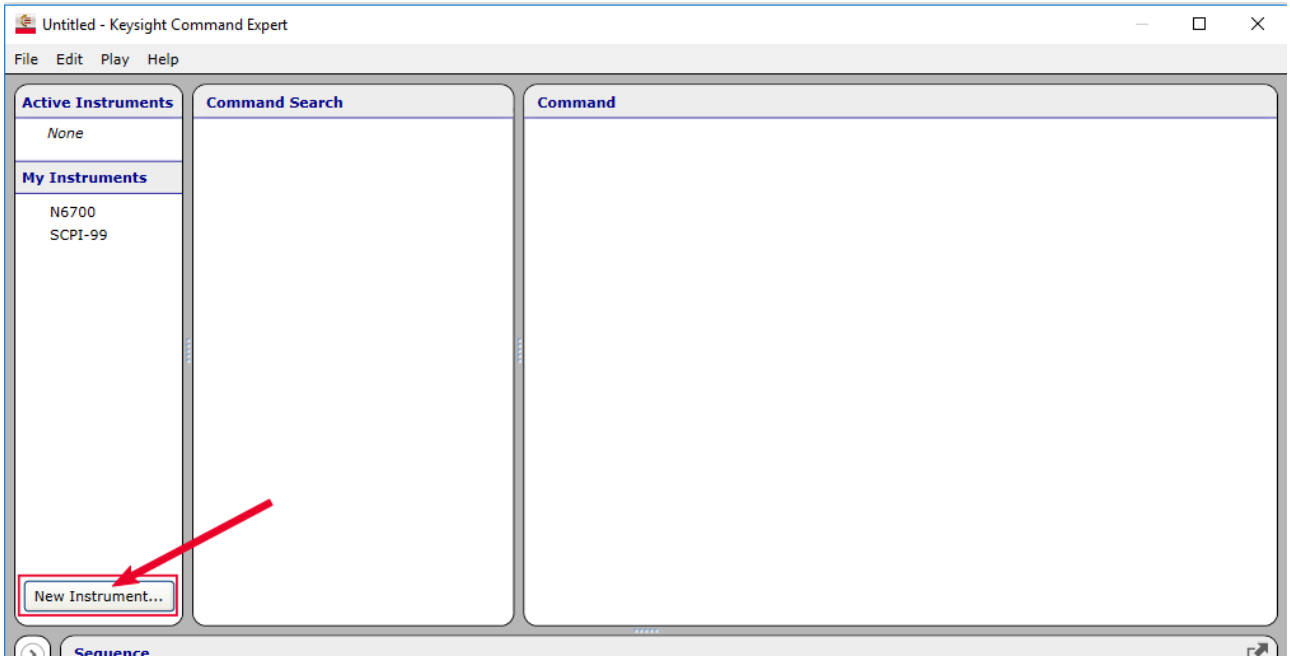
Note: Command Expert can also be launched in stand-alone mode, which **does not** allow you to export steps/commands to the TAP GUI. When using Command Expert with TAP, always launch it from the TAP GUI.

We are now ready to edit in Command Expert.

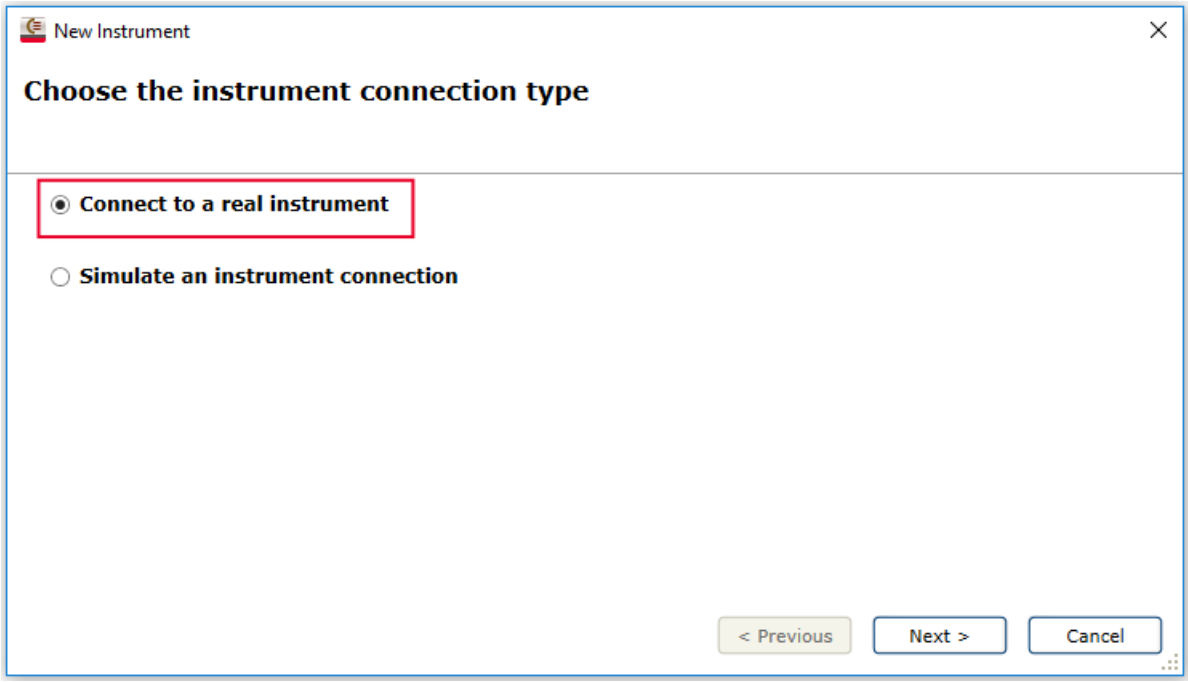
Connecting to an Instrument

In this lab we will build a simple **Command Expert sequence** to control the MXG. First we must connect to the instrument. Follow these steps:

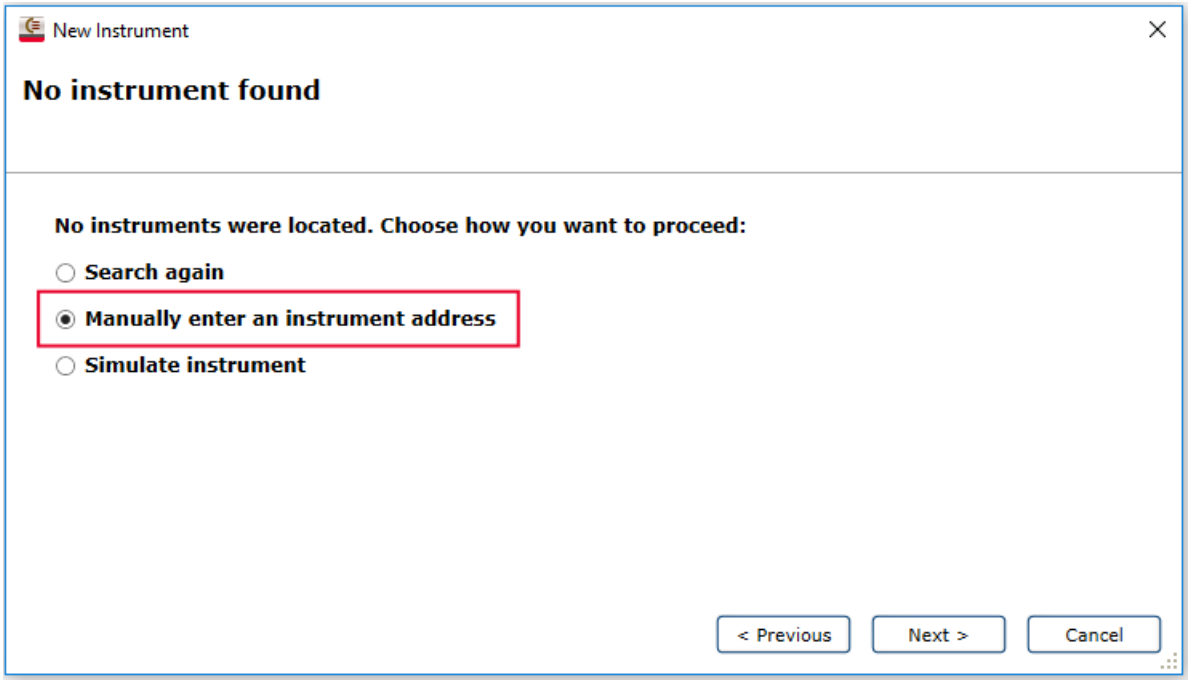
1. In Command Expert, click the **New Instrument...** button:



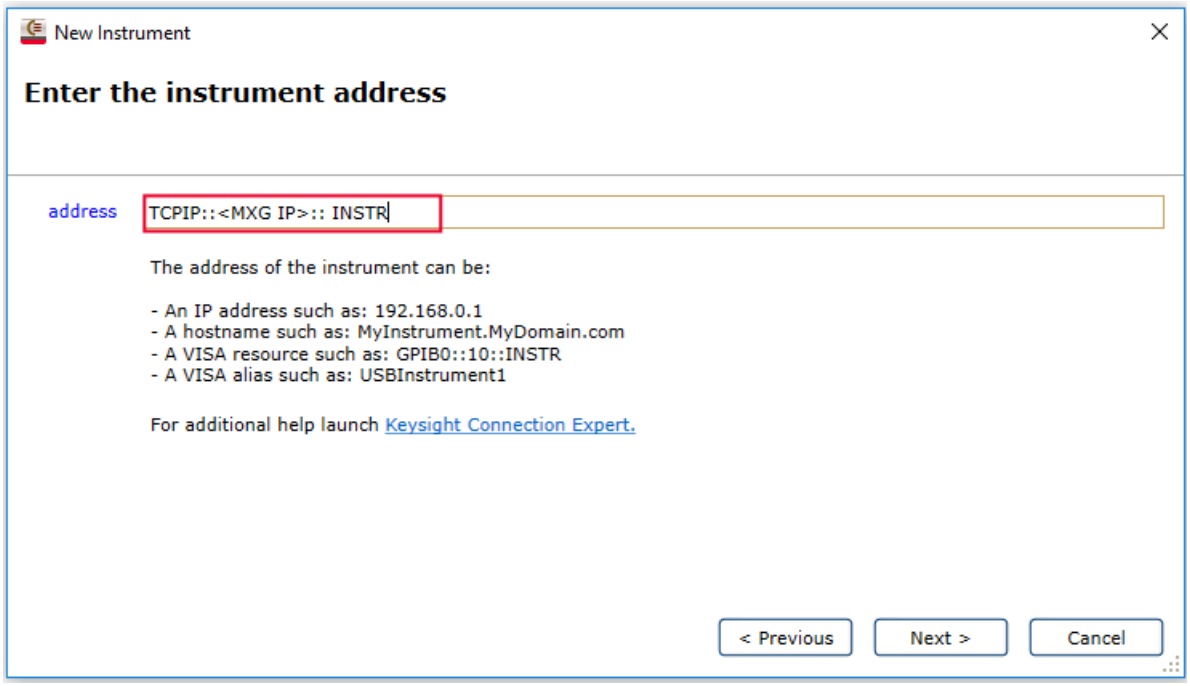
2. Connect to your MXG
   - Select **Connect to a real instrument**, then click **Next**:



   - Now click **Manually enter an instrument address**:

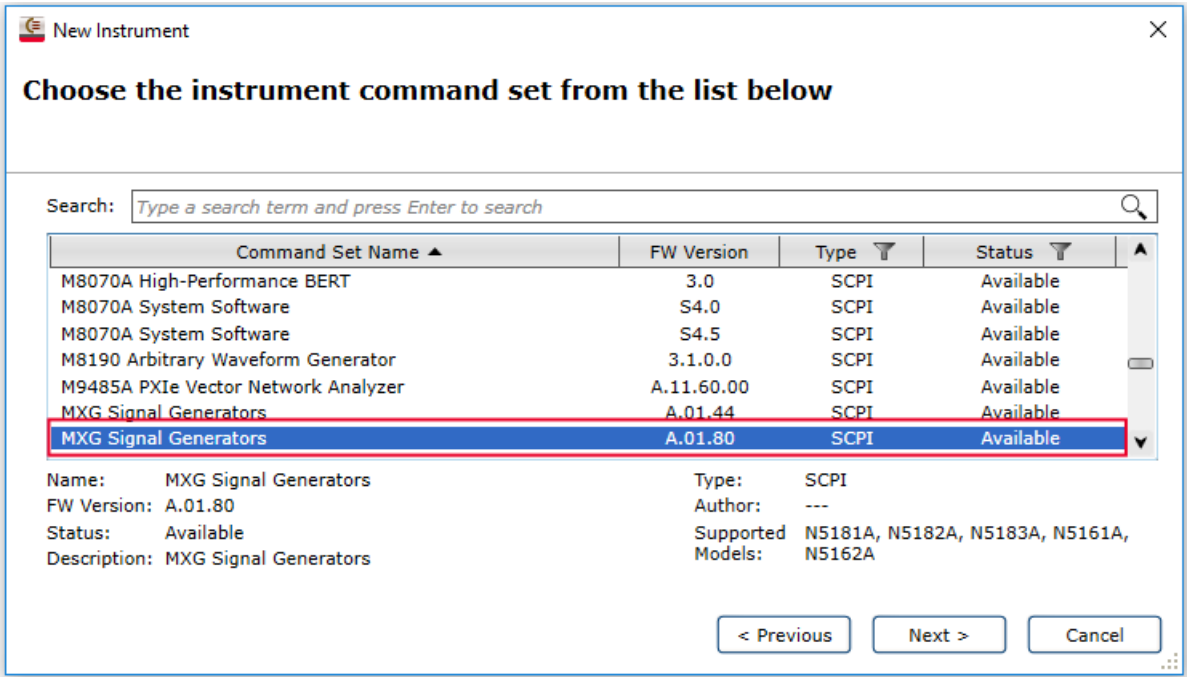


- Enter the MXG instrument's **IP address**, then click **Next**:

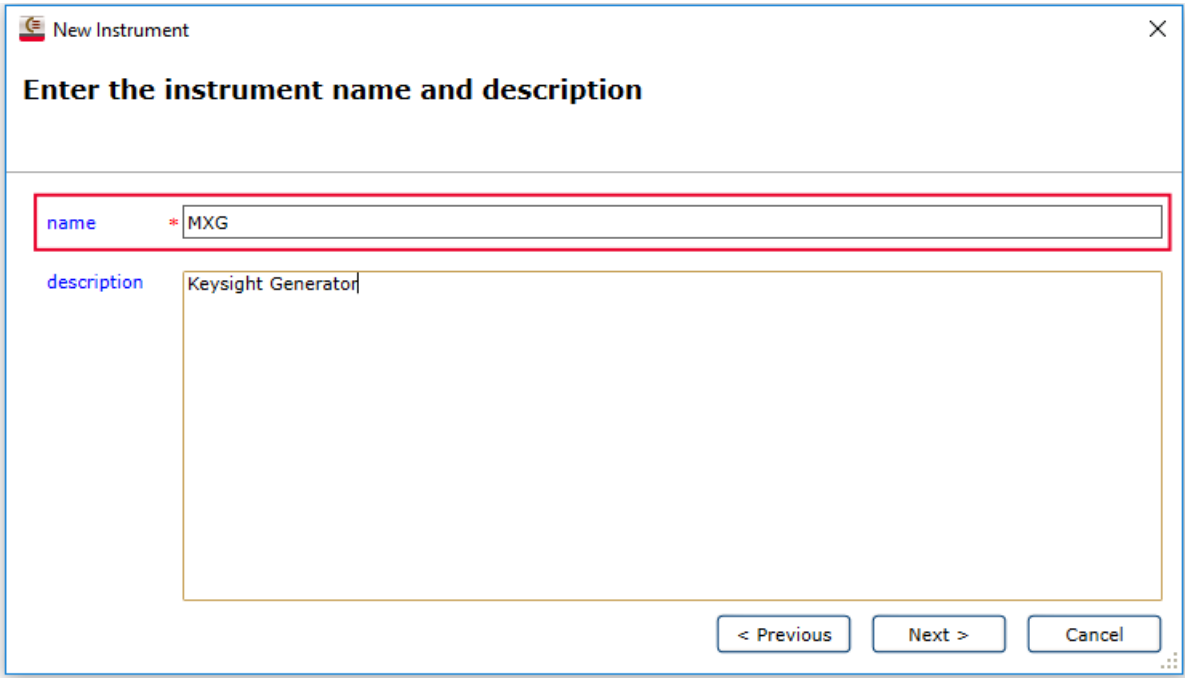


Note that you can also simulate a connection, however this is not covered in this lab.

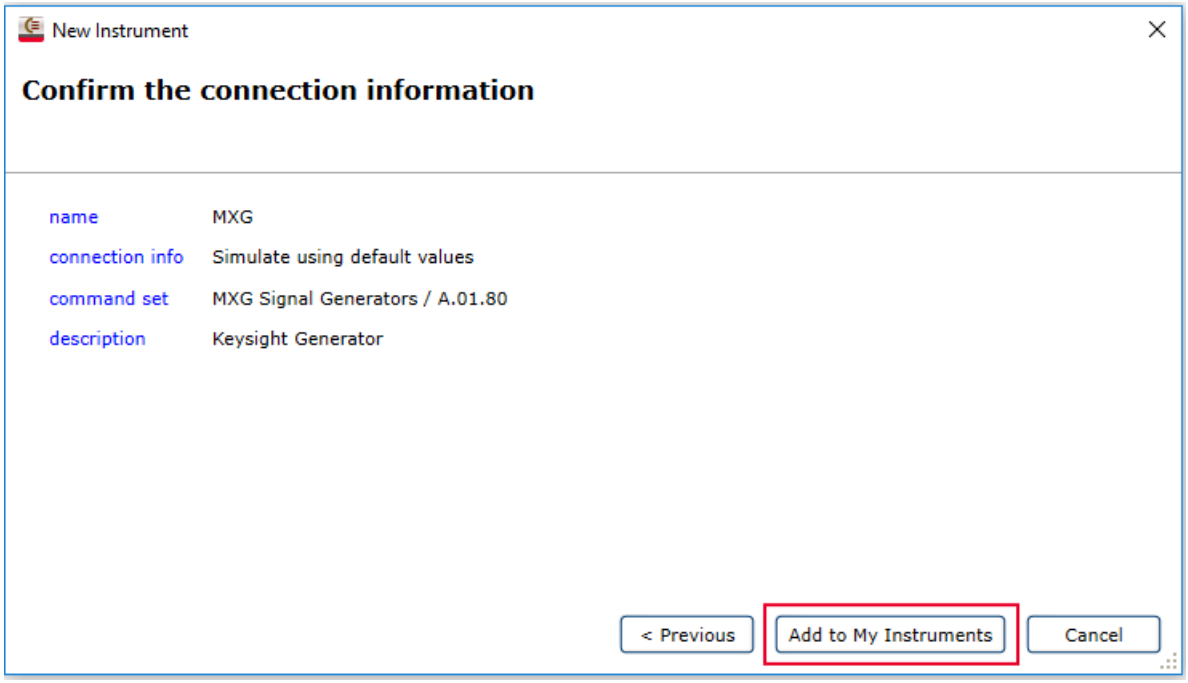
3. Select the **MXG Signal Generators** with the appropriate **FW Version** (most recent in this case) and **Type** set to **SCPI**, then click **Next**:



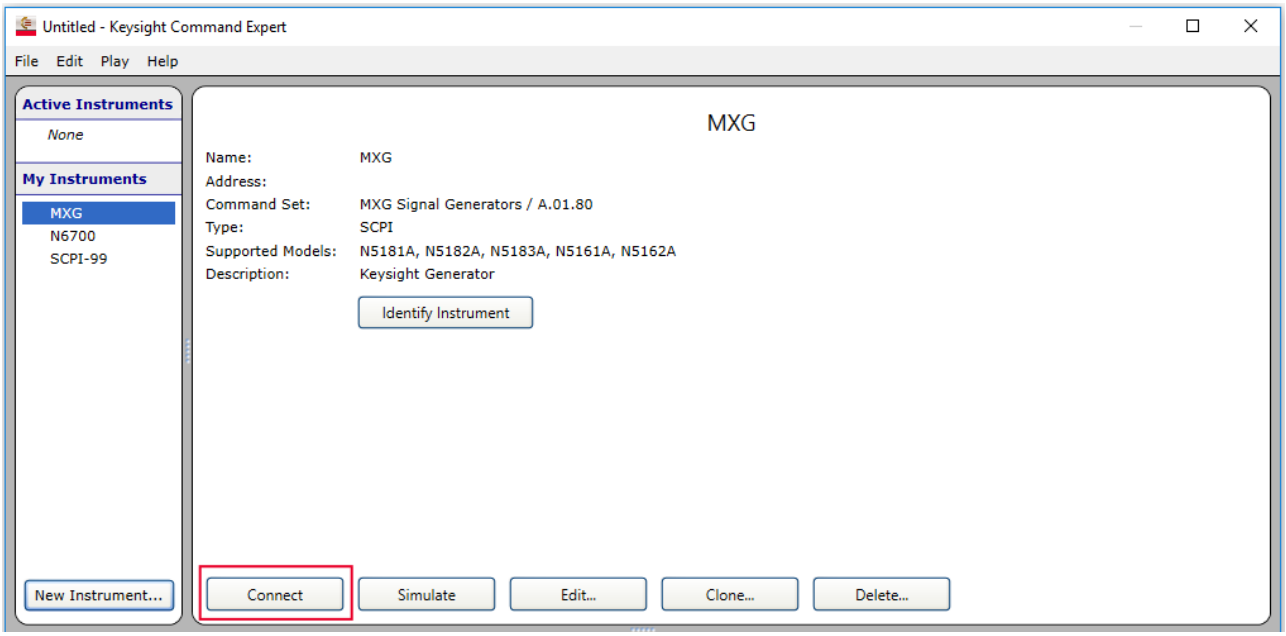
4. Enter the instrument **name** and **description** (optional), then click **Next**:



5. In the final dialog, click **Add to My Instruments** to add the instrument to Command Expert:

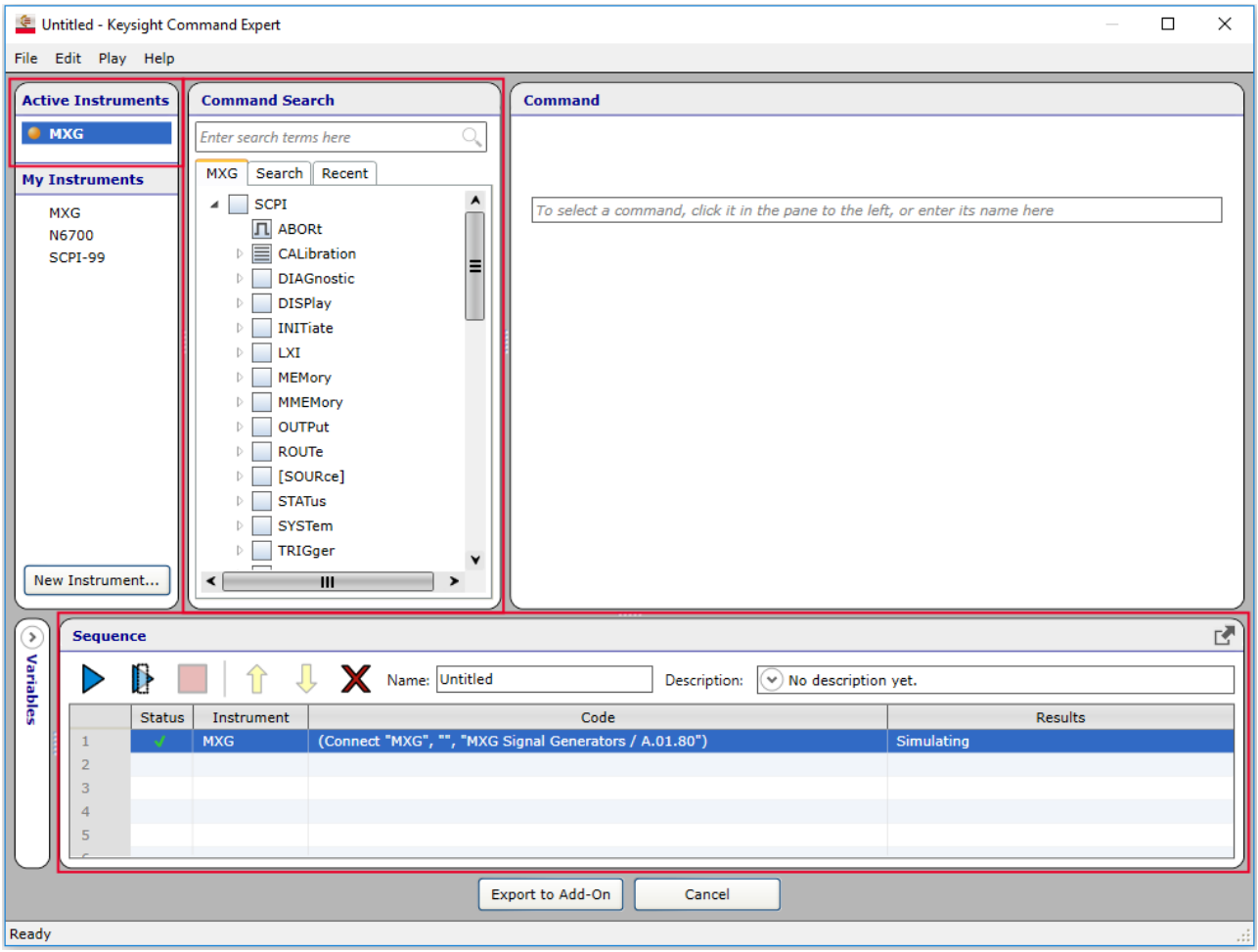


6. Now that the MXG is added to the list of instruments, click the **Connect** button:



7. Command Expert now shows:

   - A list of **Active Instruments** (top left panel)
   - A tree of **SCPI commands** in the **Command Search** panel (center section)
   - A list of **instrument commands** in the **Sequence** panel at the bottom

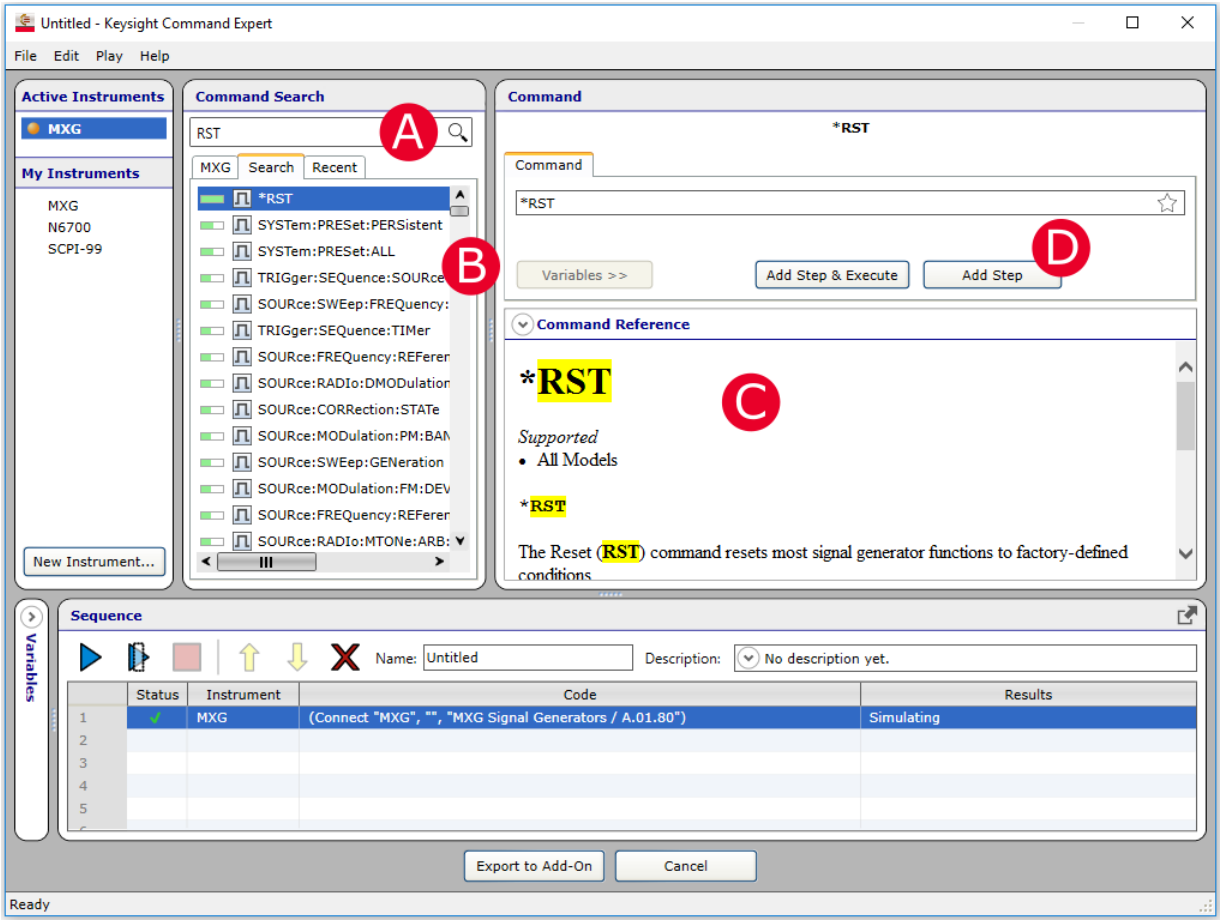


We are now ready to add SCPI commands to the sequence.

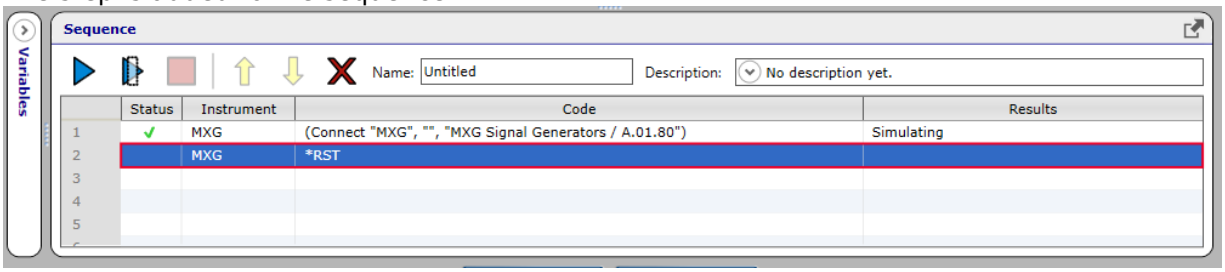
Building the Test Step Command Sequence

Now we will create a command sequence that will reset the MXG, disable RF output, set the frequency and amplitude, and then send the signal. Follow these steps:

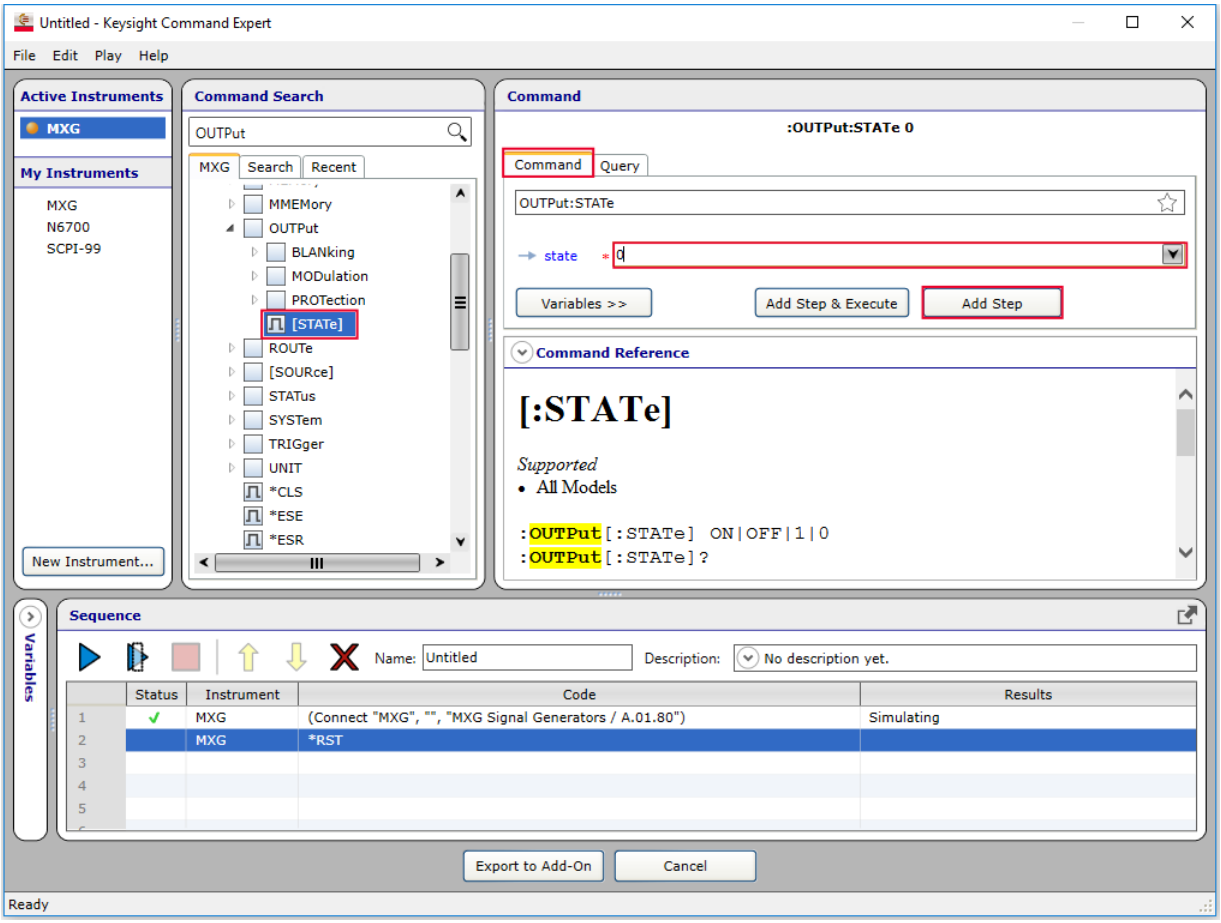
1. Add the `*RST` command, which resets most signal generator functions to factory-defined conditions:
    - In the **Command Search** panel, search (A) or navigate (B) to locate `*RST`. Note that the SCPI programming documentation for the command appears in the Command panel on the right (C).
    - Click **Add Step** to add a step to the sequence (D):



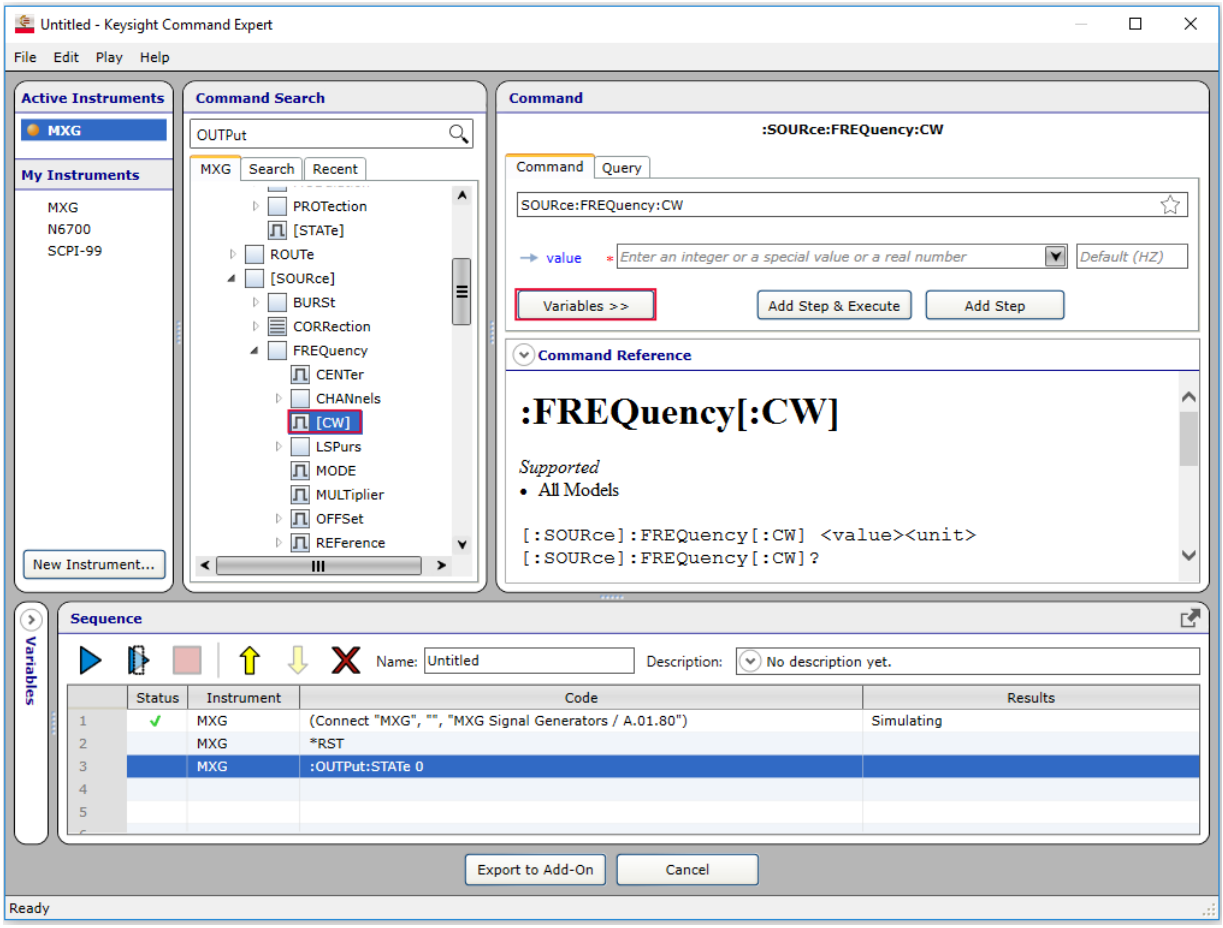
- The step is added to the sequence:



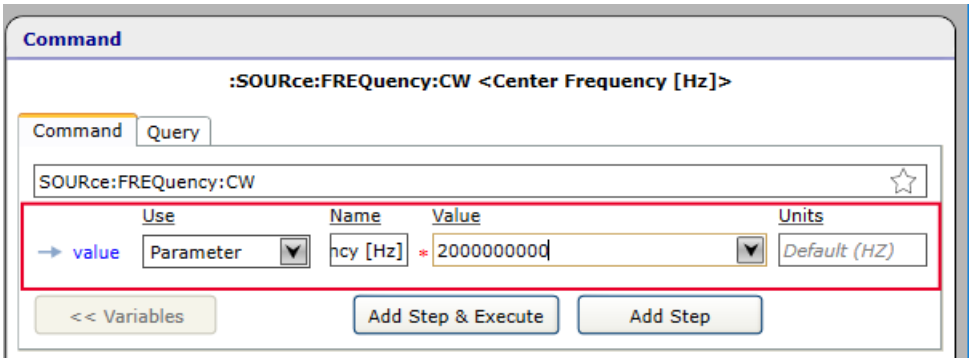
2. Add the `[STATe]` command, which enables or disables the RF output on the instrument.
   - In the Command Search panel, navigate to `OUTPut` > `[POWer]` > `[STATe]`.
   - In the Command panel, enter **0** in the **state** field.
   - Click **Add Step**.



3. Add the `[CW]` command, which sets the signal generator output frequency for the CW frequency mode.
    - In the Command Search panel, navigate to `[SOURce]` > `FREQuency` > `[CW]`.
    - In the **Command** panel, click the **Variables >>** button:

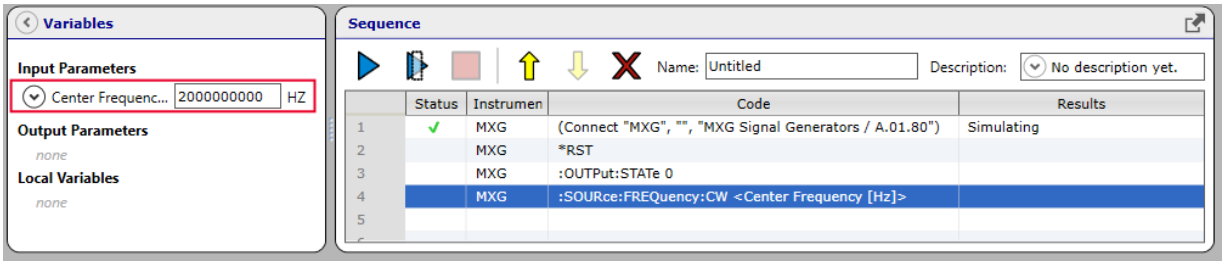


- Configure the variable as follows:
  - Use = Parameter
  - Name = Center Frequency [Hz]
  - Value = 2e9



Notice the arrow (→) next to the parameter. This indicates that this is an **input parameter**.

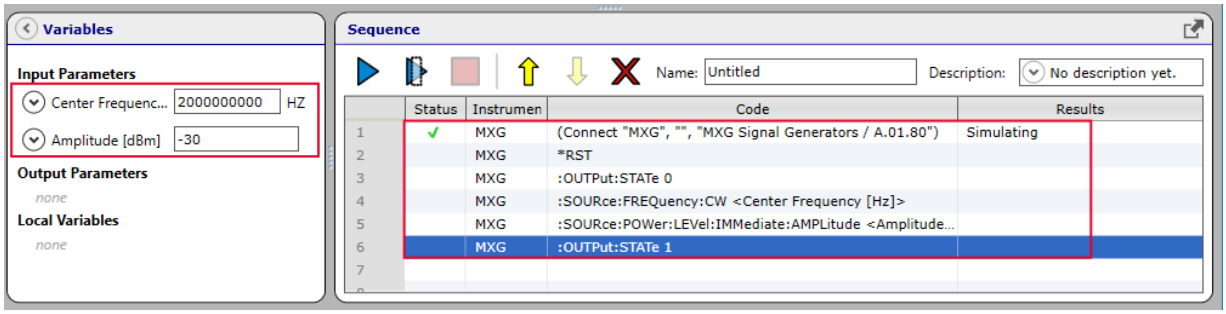
- Click **Add Step**. Now you have an input parameter as you can see in the left:



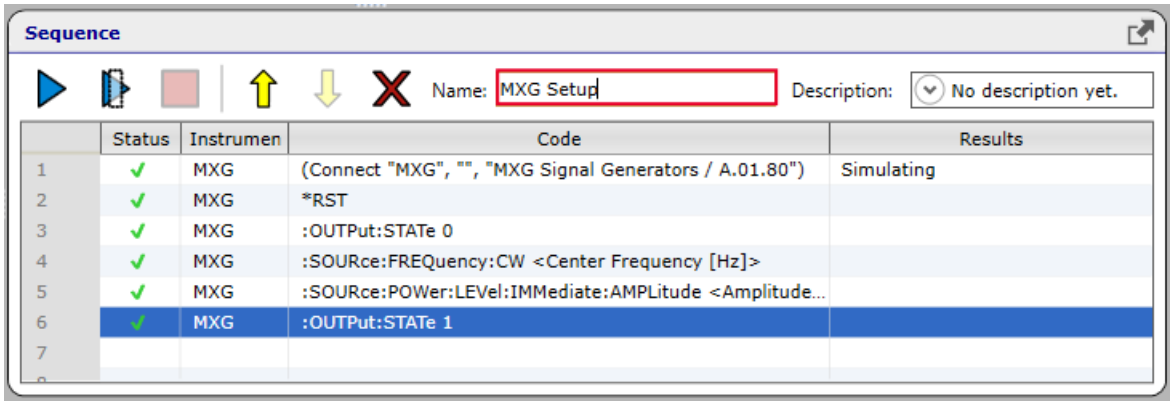
4. Add the **`[AMPlitude]`** command, which sets the RF output power:

- In the Command Search panel, navigate to `[SOURce]` > `POWer` > `[LEVel]` > `[IMMediate]` > `[AMPlitude]`.
- In the **Command** panel, click **Variables >>**. Configure the variable as follows:
  - Use = Parameter
  - Name = Amplitude [dBm]
  - Value = -30
- Click **Add Step**.

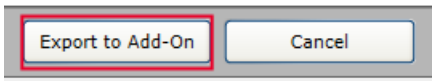
5. Add a `[STATe]` command to enable RF output:
   - In the Command Search panel, navigate to `OUTPut` > `[POWer]` > `[STATe]`. In case this doesn't exists navigate to `OUTPut` > `[STATe]`
   - In the **Command** panel, click **<< Variables**.
   - Set the **state** field to **1**, then click **Add Step**.
6. You should now see:
   - Two input parameters in the lower left panel
   - Six steps (including the Connect step) in the **Sequence** panel



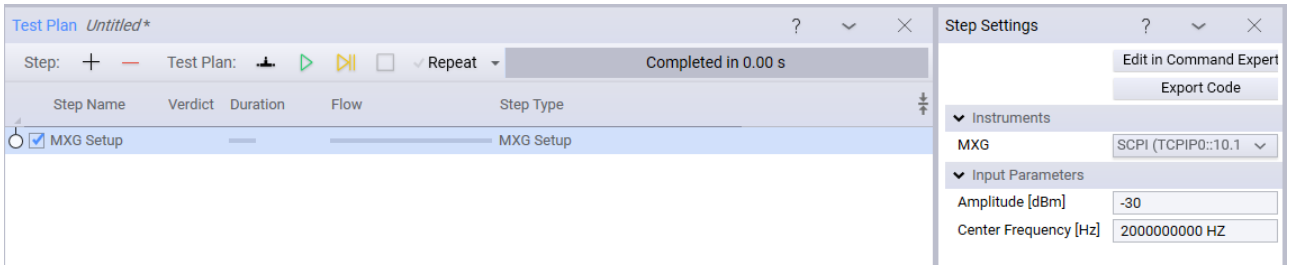
7. Click the **Play Sequence** button ( ) to run the sequence. The MXG will switch output off, set center frequency to **2GHz**, and then switch output on again.
8. In the **Sequence** panel **Name** field, enter **MXG Setup**:



9. Click the **Export to Add-On** button at the bottom of the window:

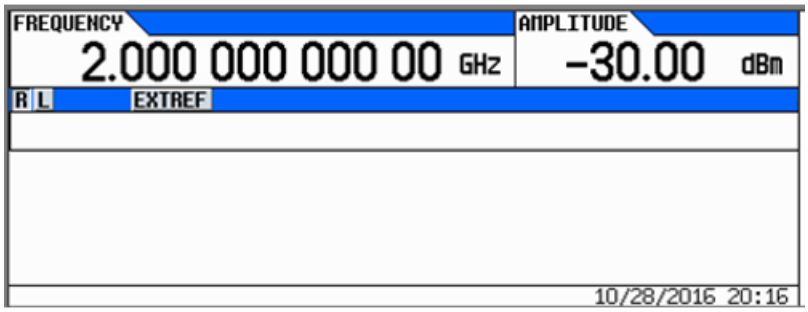


10. This returns you to the TAP GUI, where the **Command Expert** step has been renamed to **MXG Setup** and is populated with SCPI commands and related properties. Select the **MXG Setup** step to inspect its step settings:



Notice the two input parameters from the Command Expert sequence.

11. Change the parameters and click **Run**. The MXG settings change. Notice the settings on the front panel of MXG:

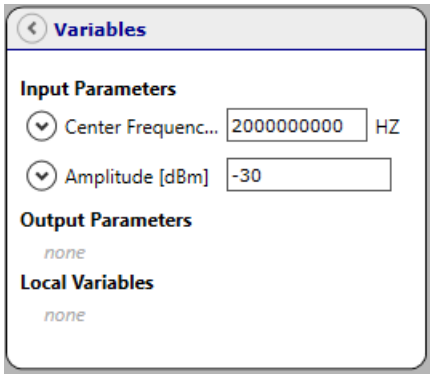


12. Set the parameters back to original values (**-30dBm/2GHz**) and click **Run** once for subsequent steps to work properly.

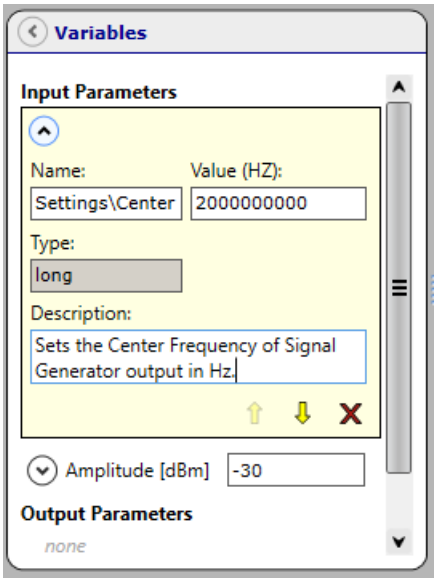
Adding More Details to the Test Step

The previously added test step does not show any description of the parameters. To add a description, follow the steps below:

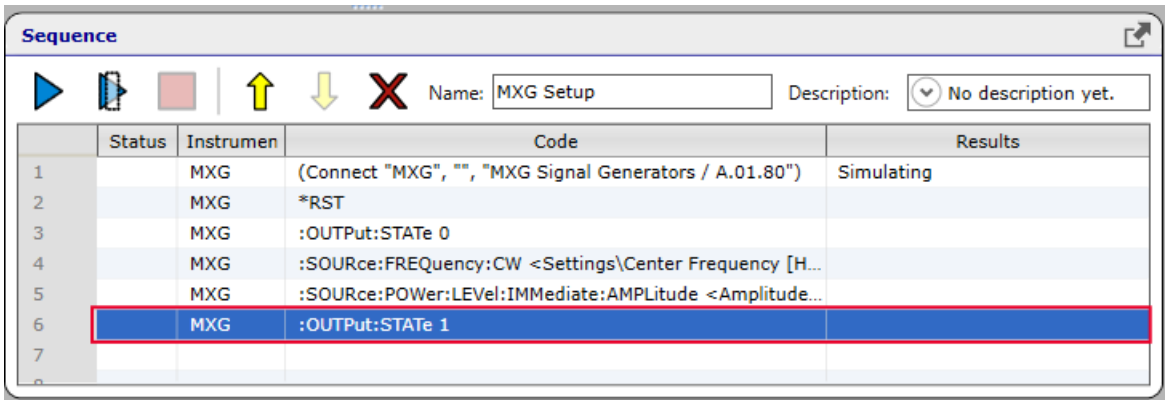
1. Select the test step and click the **Edit in Command Expert** button.
2. When **Command Expert** opens, focus on the **Variables** dialog:



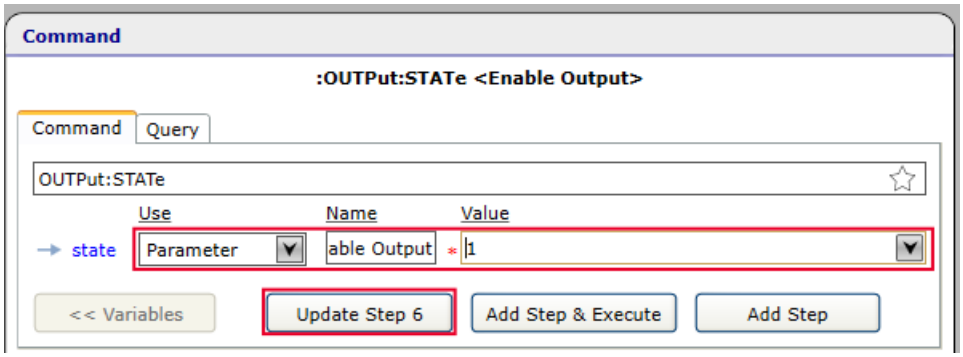
3. In this dialog:
    - Expand the **Center Frequency** parameter.
    - Rename the Center Frequency input parameter to **Settings/Center Frequency [Hz]**.
    - Enter a description.



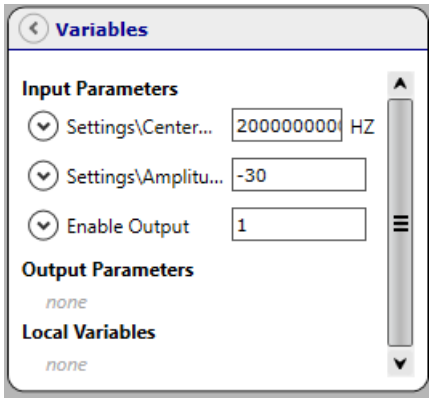
4. Similarly, expand **Amplitude**, and:
    - Rename it to **Settings/Amplitude [dBm]**.
    - Add the description **Sets the RF Output power in dBm** .
5. Next we will enable/disable RF output on the instrument from the **Test Step**:
    - Select the last step of the sequence:



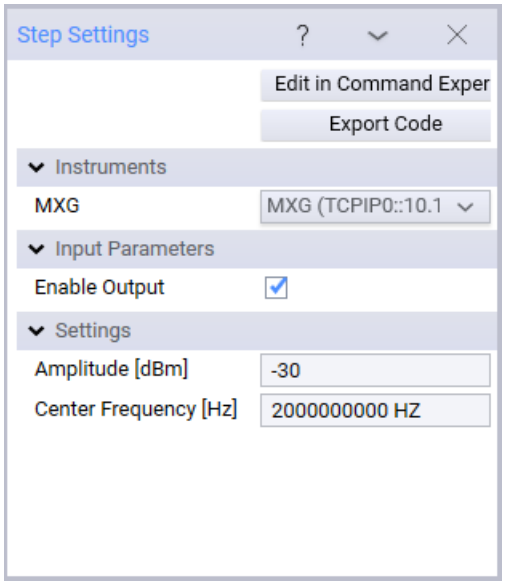
6. In the **Command** panel, click the **Variables >>** button. Configure the variable as follows:
    - Use = Parameter
    - Name = Enable Output
    - Value = 1



7. Click **Update Step 6**. The **Enable Output** parameter now appears as an **Input Parameter** in the lower left corner and has a value of 1:



8. Click **Export to Add-On**. When the TAP GUI is visible, click on the **MXG Setup** step. You will see that the step now looks more structured. You can hover the mouse over the step properties or settings to reveal additional help text:

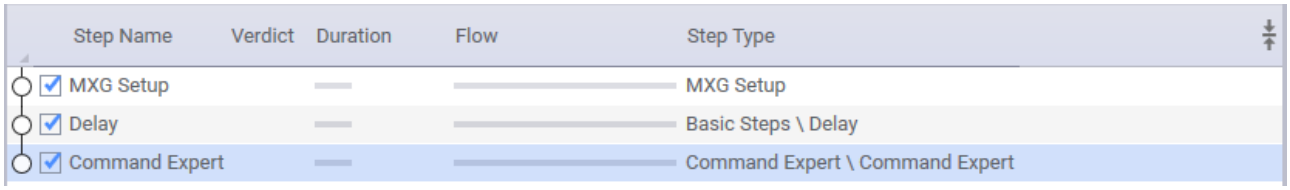


Command Expert Output Parameters in TAP

Now we will see how Command Expert output parameters can be used in TAP.

1. In the **TAP GUI**:
    - Add a **Delay** step to the test plan after the **MXG Setup** step.
    - Set the **Time Delay** of the Delay step to **1s**; this allows settling time.
    - Add a new **Command Expert test step** after the Delay step.

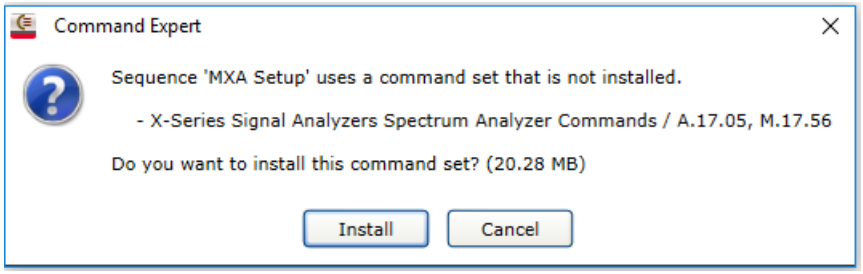
    The resulting test plan should look like this:



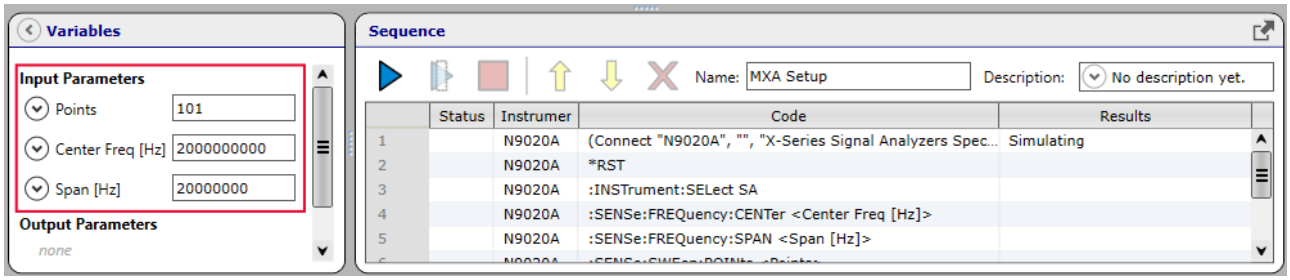
2. Select the **Command Expert** step and click **Edit in Command Expert**.

3. In the Command Expert window, select **File > Open Sequence** (or press CTRL+O) and open the **MXA Setup.iseq** file (copied earlier from USB stick to a temp folder on your PC).

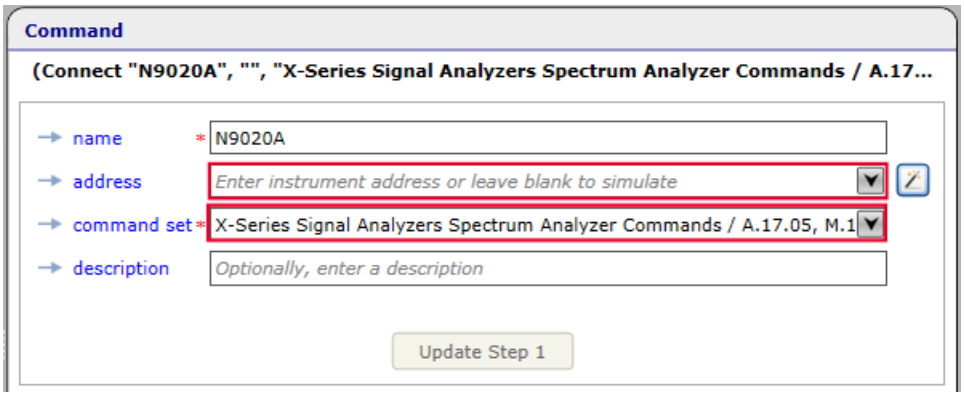
    This loads a measurement sequence for the MXA. If you see the following dialog, you do not have the command set installed. Click **Install**:



4. Notice that the sequence already contains some input parameters:



5. In the **Sequence** panel, select Step 1: **(Connect...)**
6. In the **Command** panel, make sure that the correct **address** and **command** set are used:
   - Click the  icon to set the **IP address** of MXA in your configuration.

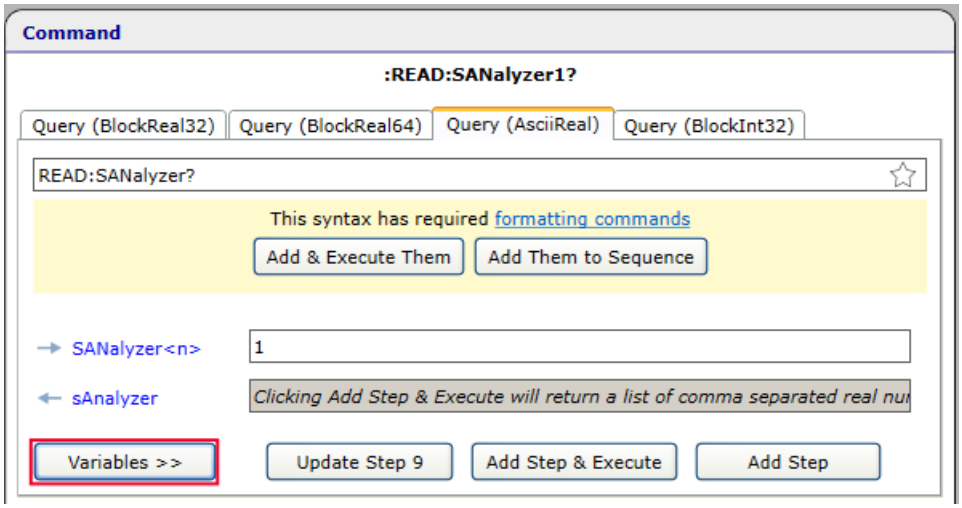


   - In the Connect Step Address dialog, select **Connect to a real instrument**, then click **Next**.
   - If you have access to an MXA, click **Manually enter an instrument address**, enter the **IP address** of the MXA in your setup, then click **Next**.
   - Click **Confirm**.
   - For command set, select the **X-Series Signal Analyzers Spectrum Analyzer Commands** with the matching (or most recent) FW version.
   - Click **Update Step 1**.

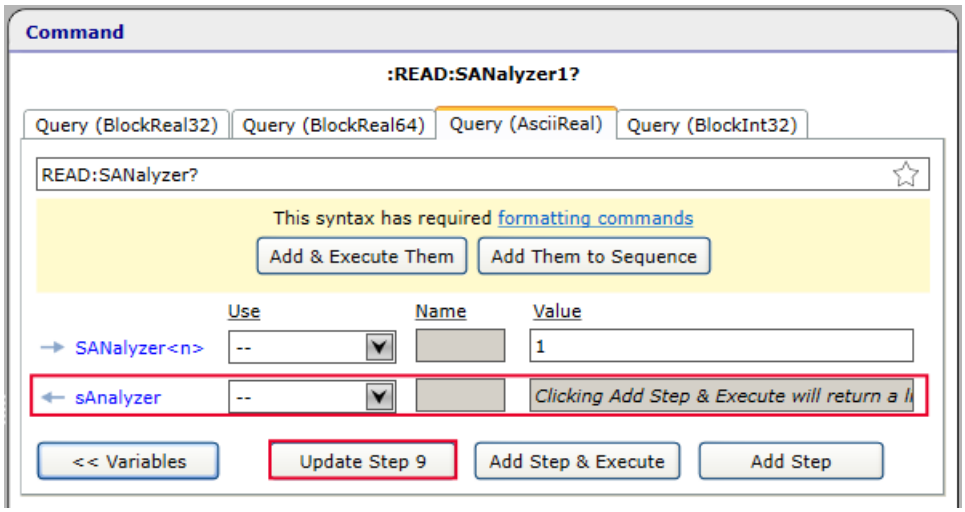
7. Run the sequence by clicking the **Play Sequence** button ().

   You will see that the last SCPI query (READ:SANalyzer?) in the sequence returned a number of values. Click on that step in the sequence and read the documentation in **Command** panel to see that the command returns points as an array of X/Y pairs containing frequency and amplitude.

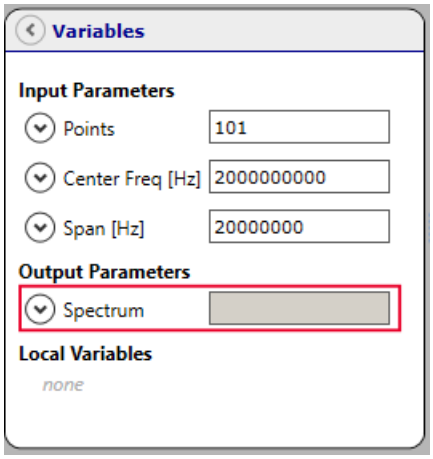
8. Now we will configure this Command Expert step to generate output values, which will be stored as results by the **TAP test step**:
   - In the Command panel, click **Variables >>**:



- Configure the **sAnalyzer** variable (notice it is an output variable, indicated by the ← icon):
  - Use = Parameter
  - Name = Spectrum

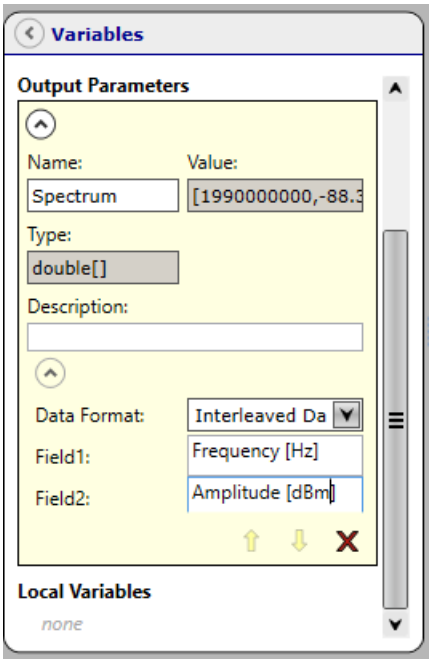


- Click **Update Step 9**. This adds an output parameter to the variables of this sequence:

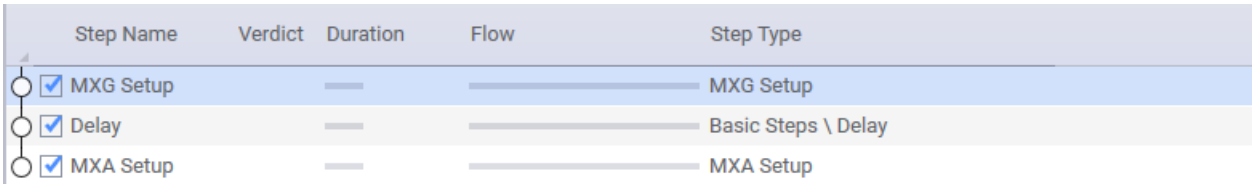


9. The values returned by this SCPI query are formatted as an array of X/Y pairs. You need to add some extra information so that TAP can make proper use of the data:

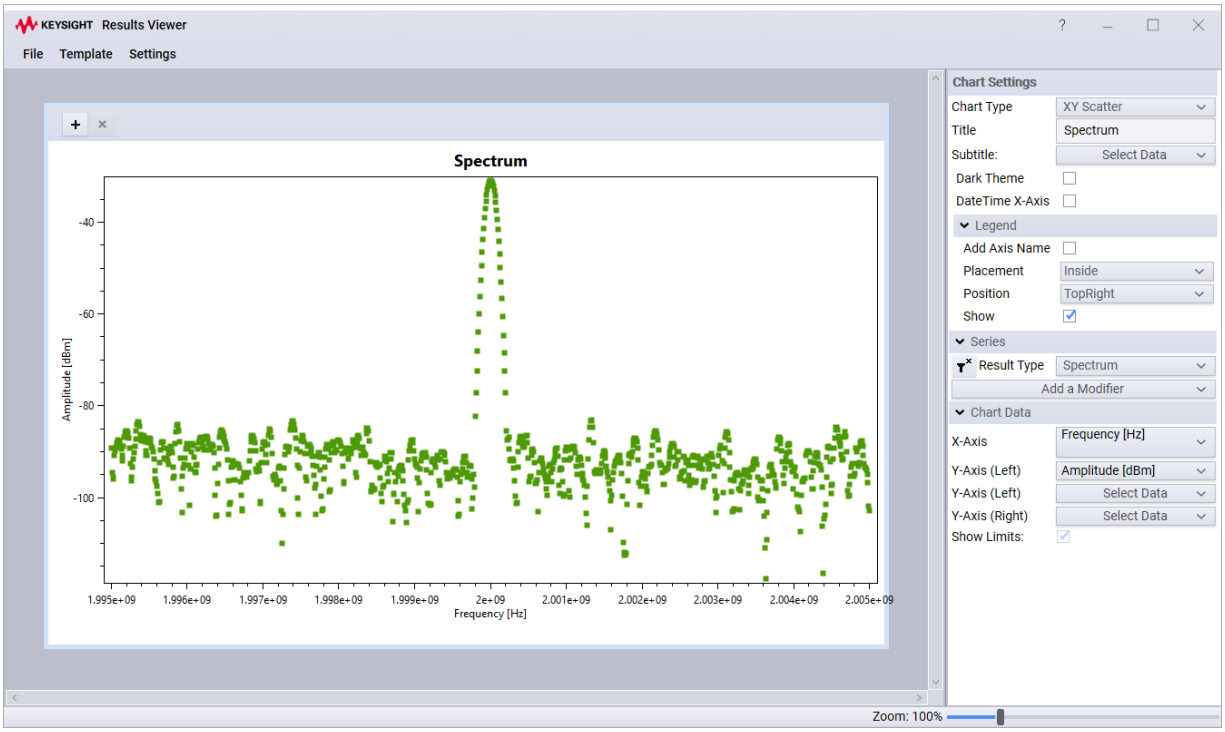
   - Expand the **Spectrum** output parameter then expand the **Data Format** group.
   - Set **Data Format** to **Interleaved DataArray 2D** as the data we get is returned pairwise.
   - In **Field1** type **Frequency [Hz]**.
   - In **Field2** type **Amplitude [dBm]**.



10. Click **Export to Add-On**. In the TAP GUI, you will see the renamed step in the **Test Plan** panel:



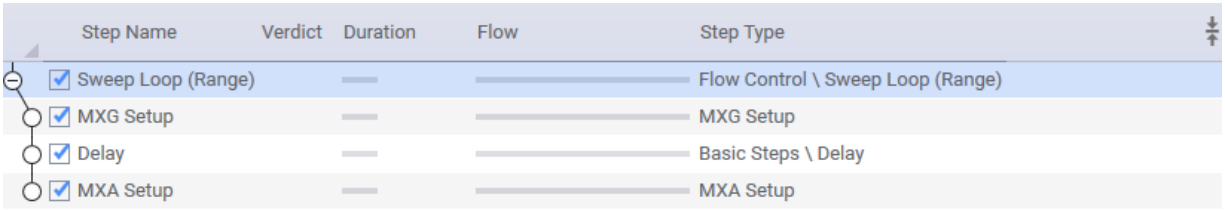
11. Click on the **MXA Setup step** and notice the Step Settings.
12. Let's look at some results:
    - Click Run, then select **Tools > Results Viewer** to see the spectrum plot. After inspecting the results, close the viewer.
    - Increase the number of points in the spectrum from **101** to **1001**.
    - Click **Run** again, then open **Results Viewer** to see the spectrum plot with more points:



- After inspecting the results, close the viewer.

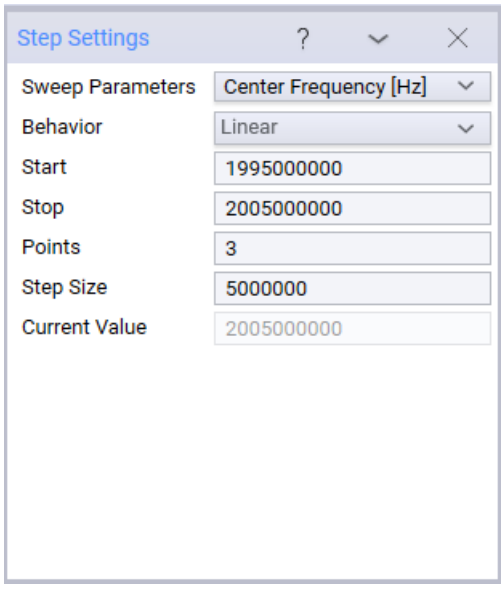
13. The **Command Expert Test Steps** in TAP can be used like any other step in TAP. Now we will make the test plan sweep over the frequencies of the MXG:

    - In the TAP GUI, add a **Sweep Loop (Range)** step below all other steps.
    - Move the three previous steps (Setup MXG, Delay, and Setup MXA) into the **Sweep Loop (Range)** step, so they become child steps of it. To do this, hold **CTRL** and click on each of the three steps. When they are selected, drag-and-drop them onto the **Sweep Loop (Range)** step. The test plan now looks like this:

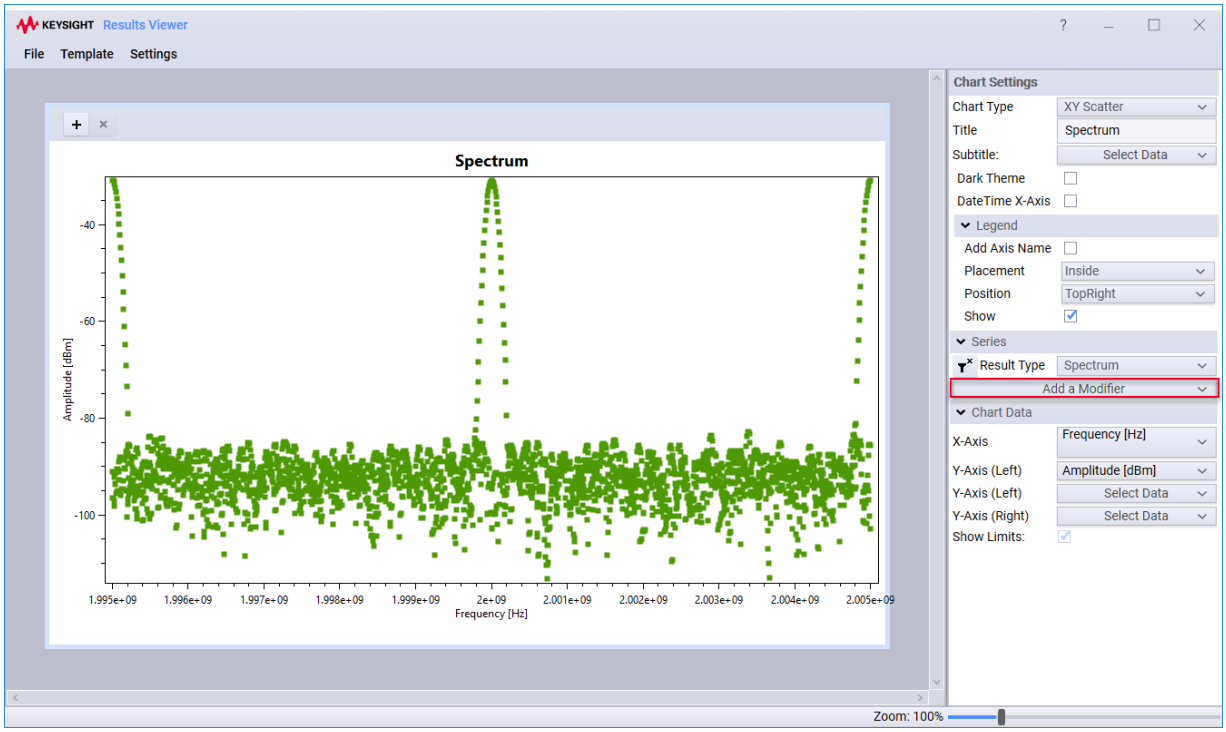


    - Click on the **Sweep Loop** step and set the Step Settings as follows:
        - Sweep Parameter — Center Frequency [Hz]
        - Behavior - Linear
        - Start — 1.995e9
        - Stop — 2.005e9
        - Step Size — 5e6

      The resulting step settings are:



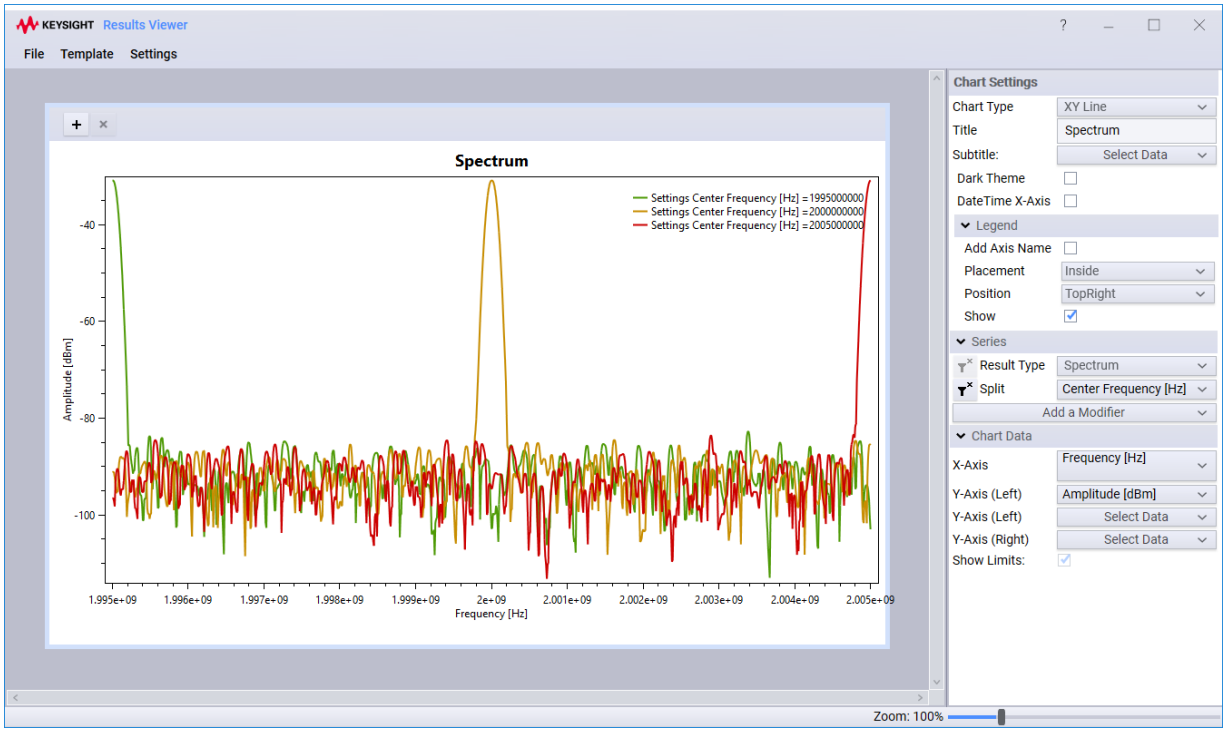
- Run the test plan, then open TAP Result Viewer to see the spectrum plot:



14. You can split the data by adding a split modifier:

    - In **Chart Settings** under **Series**, click **Add a Modifier** and select **Split**.
    - Select **Settings > Center Frequency [Hz]** for the split.

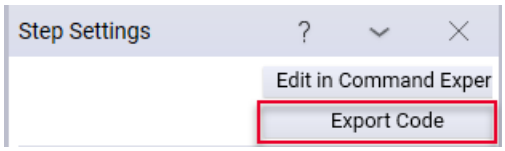
    - Set **Chart Settings > Chart Type** to **XY Line**. Now the chart looks like this:



- Close TAP Result Viewer.

15. Now that you have set up the MXG and MXA for your tests, you want to start coding to do some more advanced logic for the MXG. Programmers can get a head start by simply exporting the code, which can then be expanded to cover more specific needs:

    - Click the **Setup MXG** step.
    - In Step Settings, click **Export Code**:



    This exports the step's logic as C# source code (*.cs) to file at the location of your choosing. Save this file.

    Inspect the file's content in a text editor. Notice the similarities with data/configuration from Command Expert.

This completes the TAP Non-Programmer exercises.

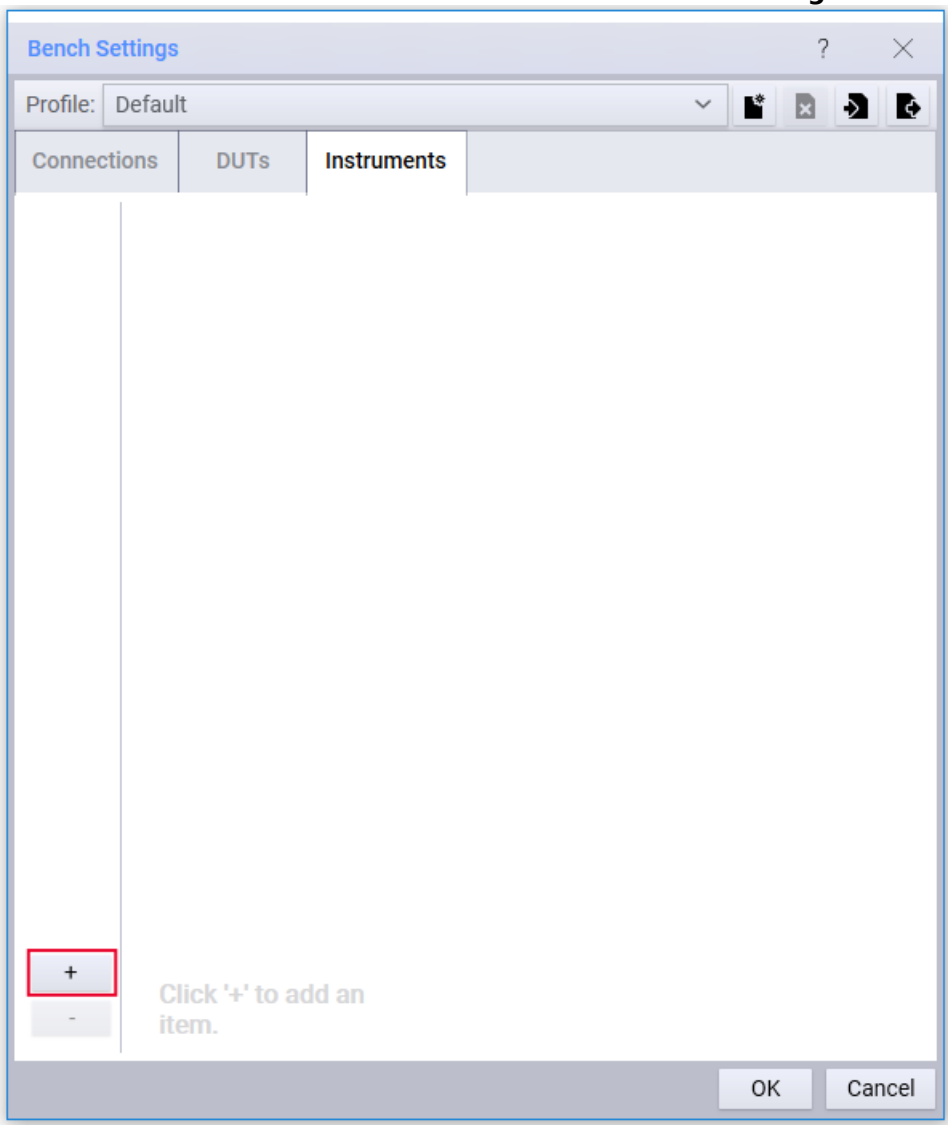
We hope you see the power of TAP and see how easy it is to interact with instruments without writing a single line of code, and that you will start using TAP for studies, projects and demonstrations! If you are curious and have time, turn the page for a few more tricks.

Bonus Exercise

Configure Instruments

This bonus exercise requires two bench instruments: an analyzer and a generator. Let’s set them up.

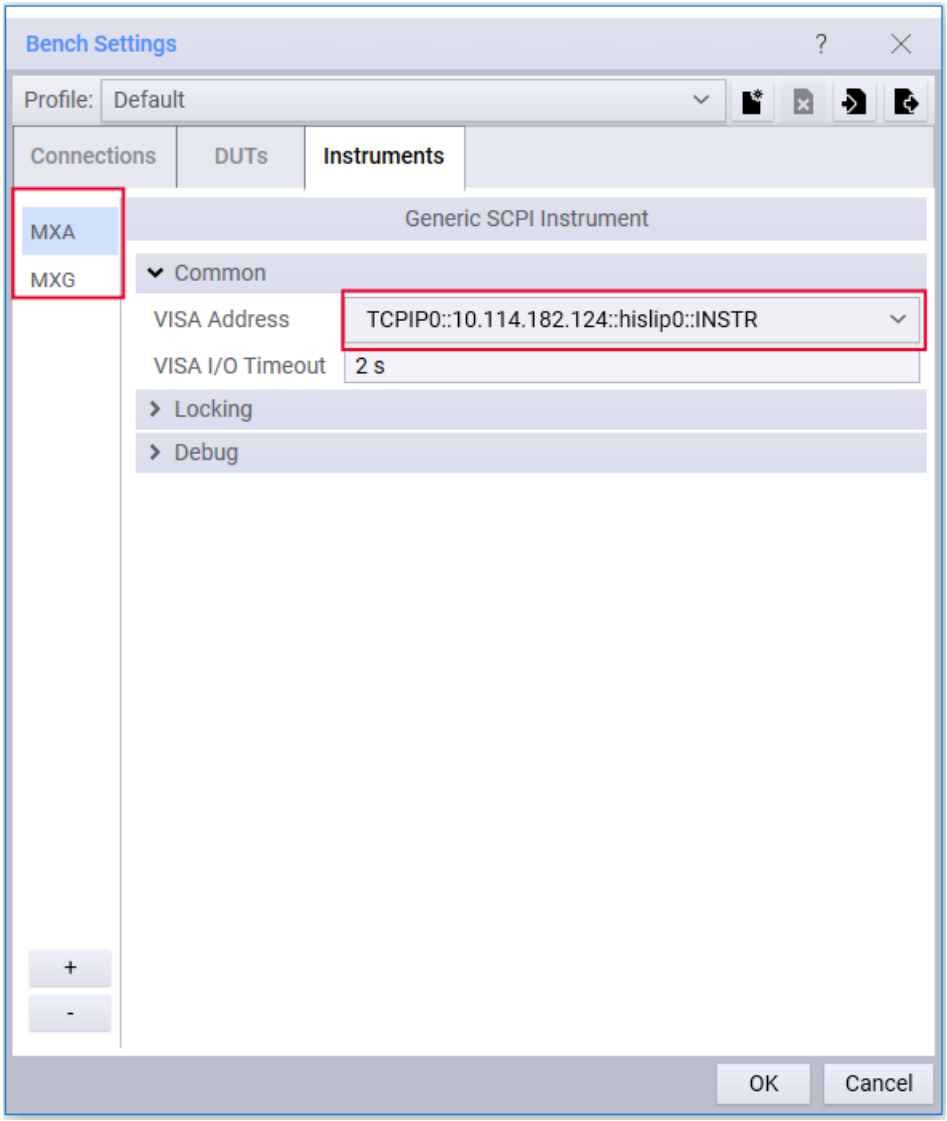
1. In the TAP GUI, select **Settings > Bench > Instruments** to open the **Bench Settings** window.
2. Click the **+** button in the lower-left corner of the **Bench Settings** window:

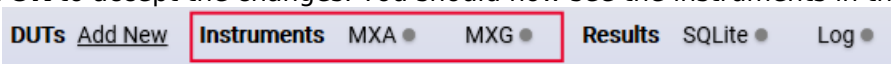


3. In the **Add New Instrument** window, which lists supported **Instruments**, add **two** instances of the **Generic SCPI Instrument** by clicking the **Add** button **twice**:

4. Close the **Add New Instrument** window.

5. In the Bench Settings window:
    - Double-click the **SCPI** and **SCPI1** instruments, and rename them **MXA** and **MXG** respectively.
    - Select each instrument and configure the **VISA Address** to:
        - MXA — `TCPIP::<MXA IP>::hislip0::INSTR`
        - MXG —`TCPIP::<MXG IP>:: INSTR`



6. Click **OK** to accept the changes. You should now see the instruments in the **Resource Bar** of TAP GUI: 

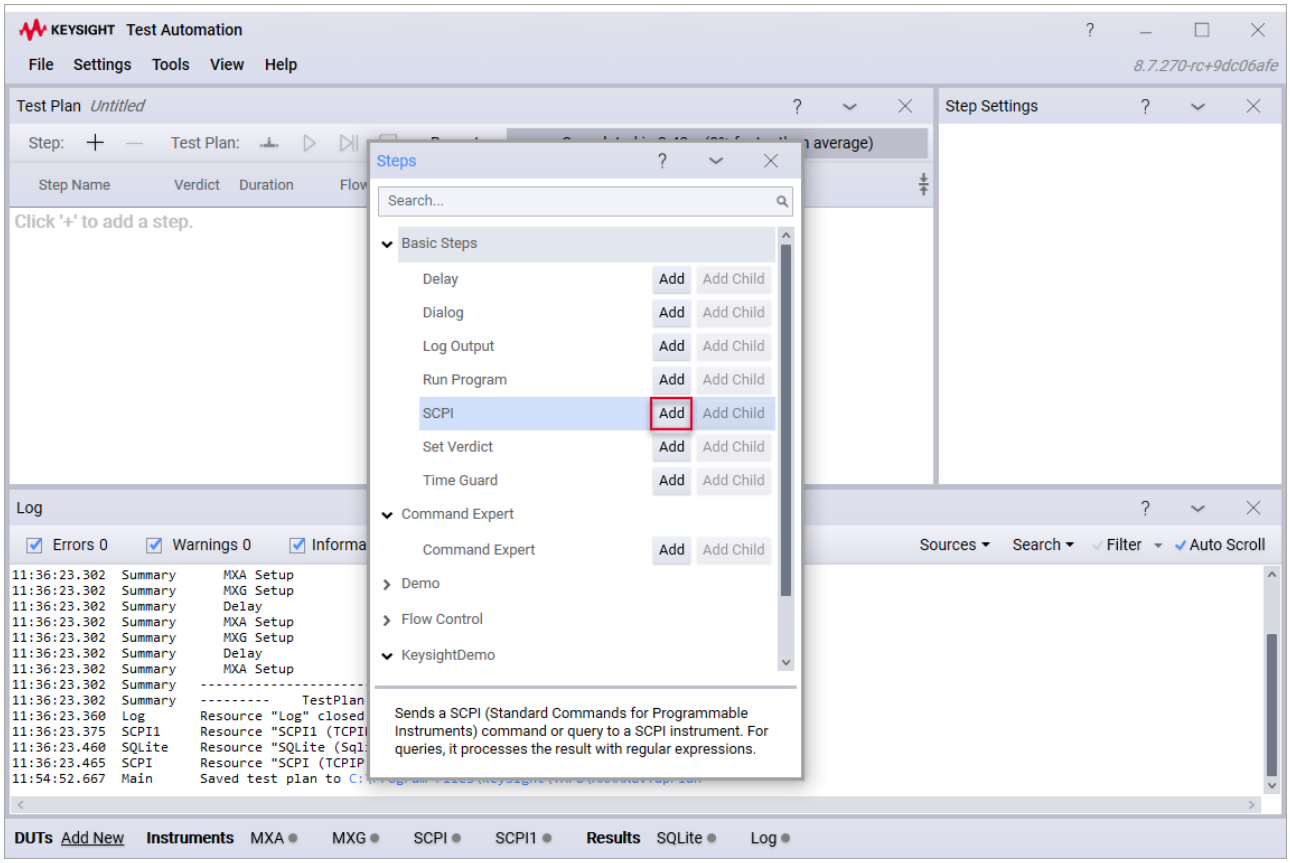
Using the SCPI Step

We will look into **the SCPI step**, a built-in TAP step for working with SCPI-based instruments. This step:

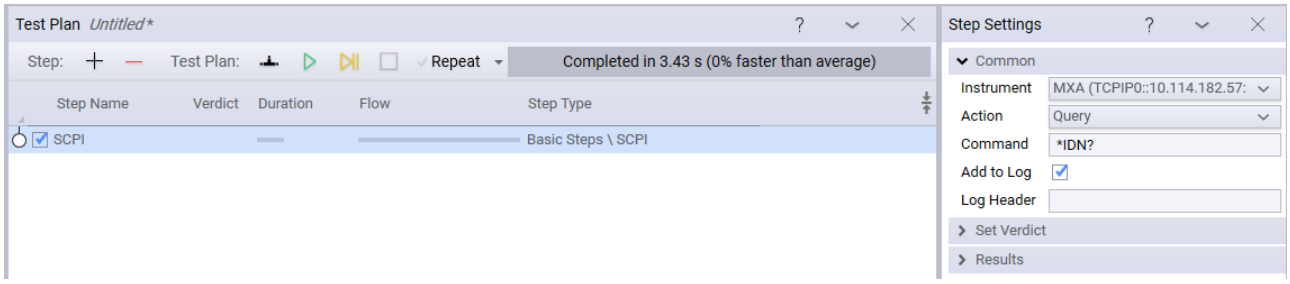
- Lets you send a single **SCPI command** or **query** to any SCPI instrument, thus enabling you to perform basic SCPI communication without doing any programming.
- Enables you to interpret the response of a SCPI query and set the test plan verdict based on that
- Provides additional features that can be used to generate a verdict and results based on text matching rules

Using the Step

1. In the TAP GUI, click **File > New** to create a new test plan.
2. Click the **+** button, then click **Add** next to the **SCPI** step from the **Basic Steps** category. Close the dialog.



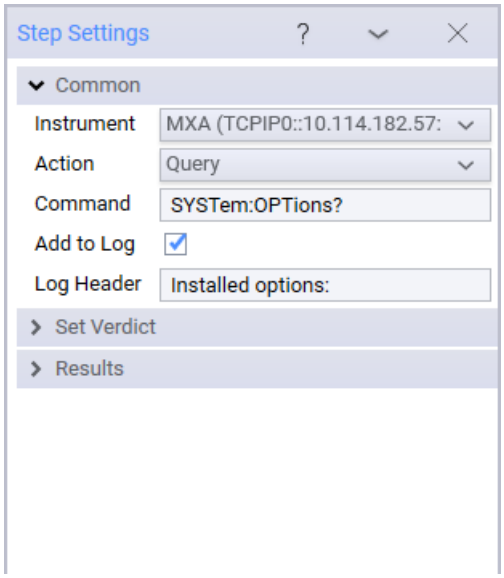
3. Click on the **SCPI step** in the **Test Plan** panel to see the **Step Settings**:



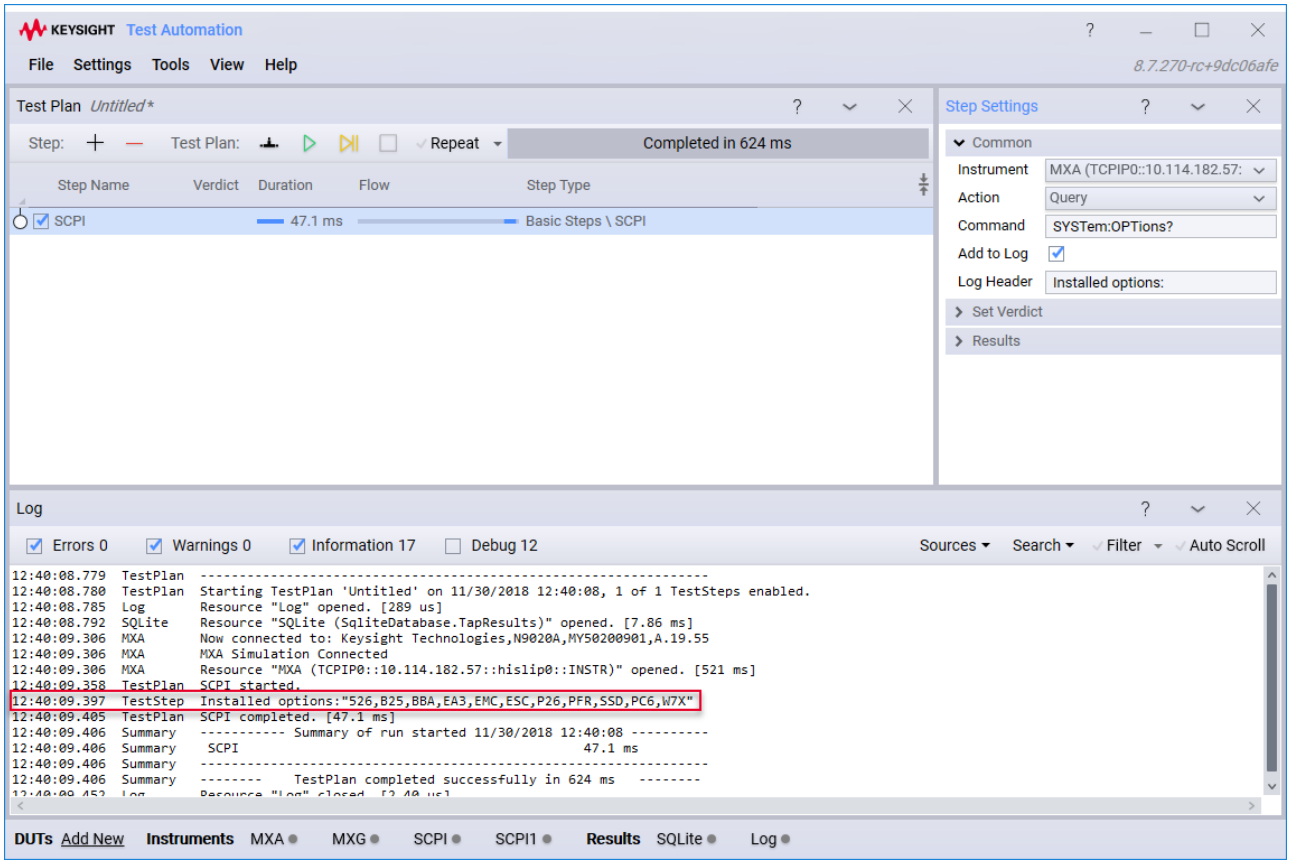
4. Set the settings as follows:

   - Instrument — Select **MXA**
   - Action — Select **Query**
   - Command — Copy and paste **SYSTem:OPTions?**
   - Add to Log — **Enable**
   - Log Header — Copy and paste **Installed options**:

   The resulting step settings are:



5. Click **Run**. The log should show a list of options installed on the MXA:

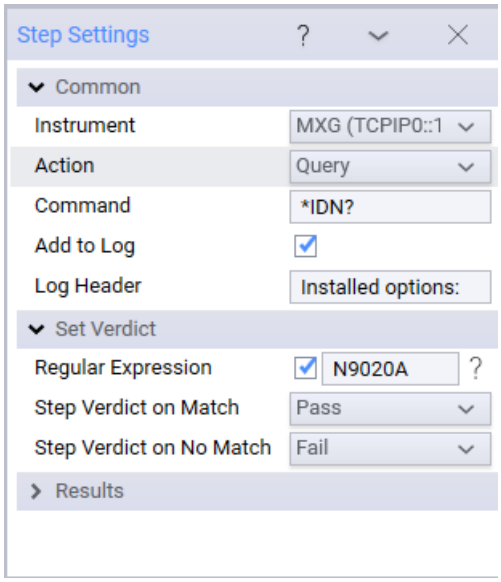


Setting a Verdict

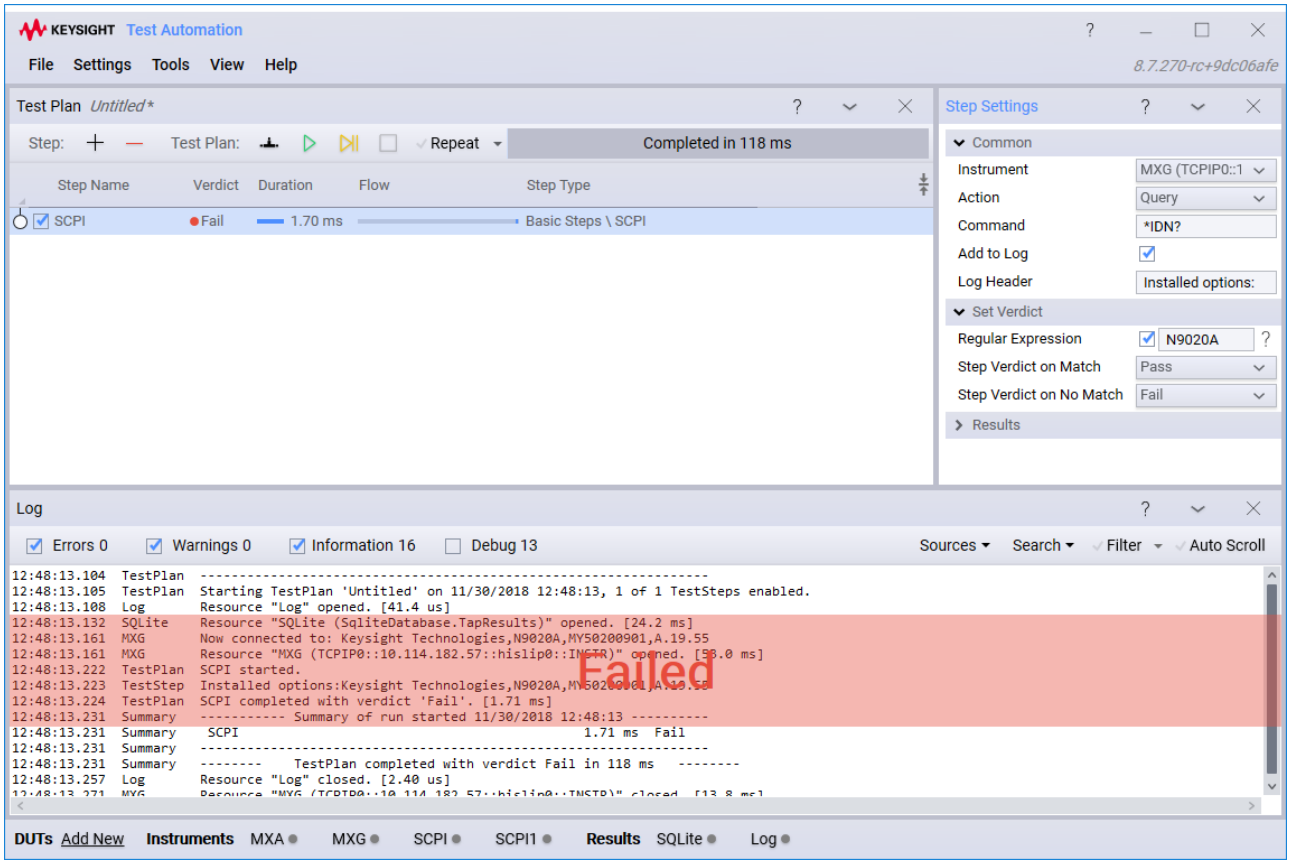
1. In the **Step Settings** panel for the **SCPI** test step, expand the **Set Verdict** group. This reveals options that can be used to test the output of a SCPI query. Configure the step as follows:

   - Instrument — Select **MXG**
   - Action — Select **Query**
   - Command — Copy and paste `*IDN?`
   - Add to Log — **Enable**
   - Regular Expression — **Enable** it and enter **N9020A**

   The resulting settings are:

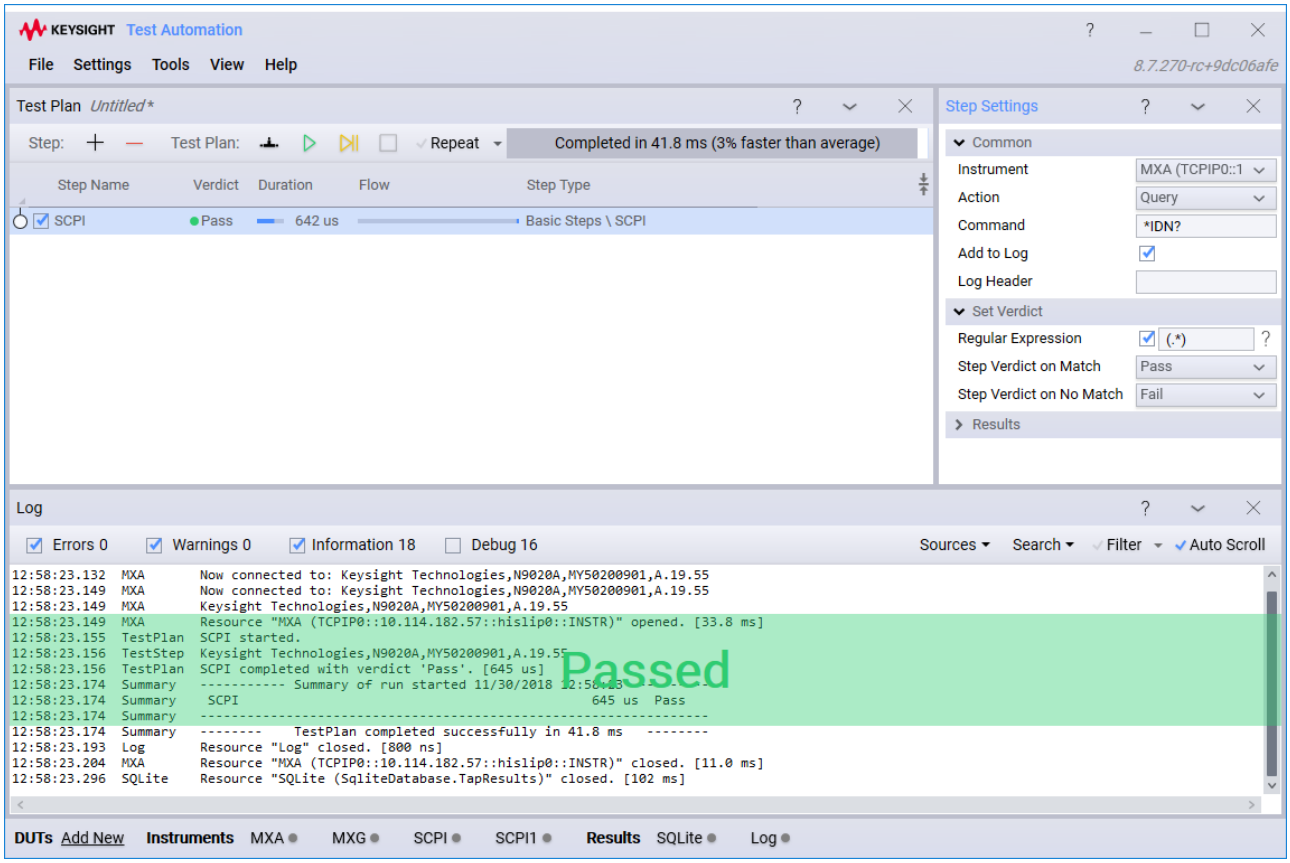


2. Click **Run**. While the test is running, you see a large red **Failed** message in the log showing that the SCPI query resulted in a Fail verdict:



This happens because in the previous step we configured the step to run for MXG. Since the verdict is set to Pass, if the query response matches with the model number of the MXA, the verdict is set to fail.

3. Now, in the SCPI step settings, change the instrument to an MXA and rerun the test plan. You should see a **Passed** verdict as shown below:


KEYSIGHT Test Automation
File Settings Tools View Help
8.7.270-rc+9dc06afe
Test Plan Untitled*
Step: Test Plan: Repeat
Completed in 41.8 ms (3% faster than average)
Step Name Verdict Duration Flow Step Type
SCPI Pass 642 us Basic Steps \ SCPI
Step Settings
Common
Instrument MXA (TCPIP0::1
Action Query
Command *IDN?
Add to Log
Log Header
Set Verdict
Regular Expression (.*)
Step Verdict on Match Pass
Step Verdict on No Match Fail
Results
Log
Errors 0 Warnings 0 Information 18 Debug 16
Sources Search Filter Auto Scroll
12:58:23.132 MXA Now connected to: Keysight Technologies,N9020A,MY50200901,A.19.55
12:58:23.149 MXA Now connected to: Keysight Technologies,N9020A,MY50200901,A.19.55
12:58:23.149 MXA Keysight Technologies,N9020A,MY50200901,A.19.55
12:58:23.149 MXA Resource "MXA (TCPIP0::10.114.182.57::hislip0::INSTR)" opened. [33.8 ms]
12:58:23.155 TestPlan SCPI started.
12:58:23.156 TestStep Keysight Technologies,N9020A,MY50200901,A.19.55
12:58:23.156 TestPlan SCPI completed with verdict 'Pass'. [645 us]
12:58:23.174 Summary ---------- Summary of run started 11/30/2018 12:58:23 ----------
12:58:23.174 Summary SCPI 645 us Pass
12:58:23.174 Summary ----------
12:58:23.174 Summary -------- TestPlan completed successfully in 41.8 ms --------
12:58:23.193 Log Resource "Log" closed. [800 ns]
12:58:23.204 MXA Resource "MXA (TCPIP0::10.114.182.57::hislip0::INSTR)" closed. [11.0 ms]
12:58:23.296 SQLite Resource "SQLite (SqliteDatabase.TapResults)" closed. [102 ms]
Passed
DUTs Add New Instruments MXA MXG SCPI SCPI1 Results SQLite Log