How to Run the Example

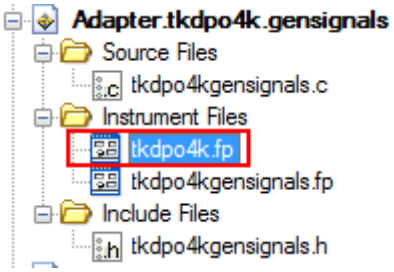
1. Preparation

1. Install LabWindows™/CVI™.
2. Install NI Measurement & Automation Explorer (MAX), IVI Compliance Package, and NI-VISA from the NI Device Drivers.
3. Install the following instrument drivers:
   Tektronix tkdpo4k IVI Driver
   Keysight hp34401a IVI Driver
   Fluke fl2638 CVI PnP Driver

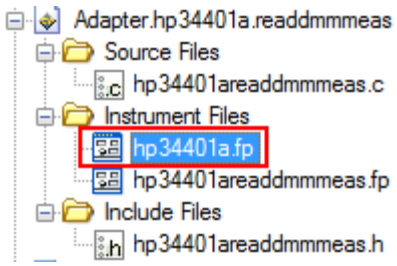
2. Build Example

Note: The projects may already include the needed `fp` or `lib` files. However, the path may be not correct because the location of `fp` and `lib` files on your computer may vary. In such case, you just need to delete them and then re-add the `fp` or `lib` files to the correct location.

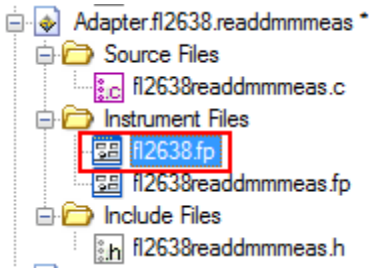
1. Open `CHAL.cws`. This workspace includes all the projects you need to build.
2. Add `tkdpo4k.fp` to project Adapter.tkdpo4k.gensignals.



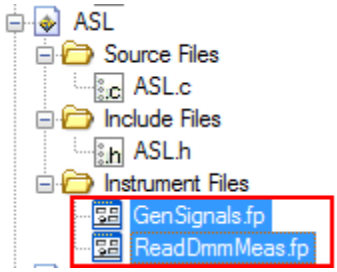
3. Add `hp34401a.fp` to project Adapter.hp34401a.readdmmmeas.



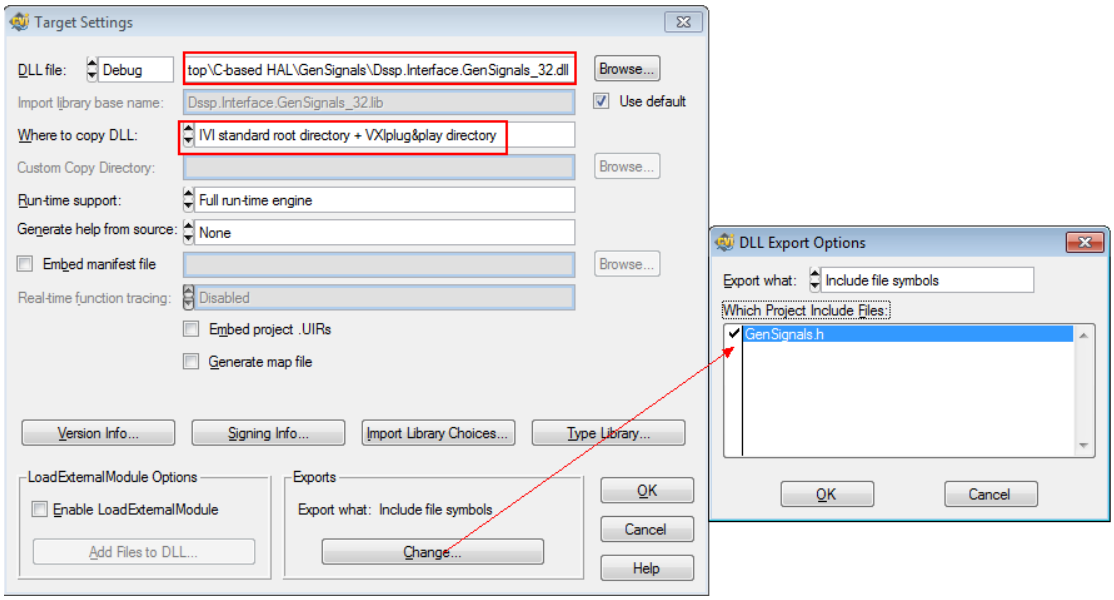
4. Add `fl2638.fp` to project Adapter.fl2638.readdmmmeas.



5. Add `GenSignals.fp` and `ReadDmmMeas.fp` to project ASL.



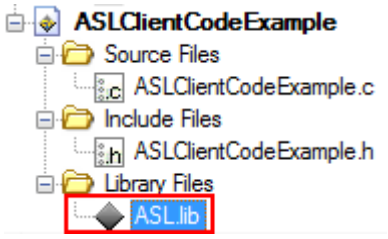
6. Make sure the build target of each project (except ASLClientCodeExample) is **Dynamic Link Library** *(***Build»Target Type***)* and the Target Settings *(***Build»Target Settings***)* is identical to the following image.
   Note: Right click on each project and select **»Set Active Project** before attempting to configure each project's target settings.



The following table lists the DLL name of each project.

Project	**DLL Name (Change 32 to 64 if building a 64-bit DLL)**
GenSignals	`Dssp.Interface.GenSignals_32.dll`
ReadDmmMeas	`Dssp.Interface.ReadDmmMeas_32.dll`
Adapter.tkdpo4k.gensignals	`adapter.tkdpo4k.gensignals_32.dll`
Adapter.hp34401a.readdmmmeas	`adapter.hp34401a.readdmmmeas_32.dll`
Adapter.fl2638.readdmmmeas	`adapter.fl2638.readdmmmeas_32.dll`
ASL	`ASL.dll`

7. From the top down, set each project as the active project and build it (except ASLClientCodeExample). You can also use batch build *(***Build»Batch Build***)* to build them all at once.
8. Add `ASL.lib` to project ASLClientCodeExample. For 64-bit version of this exercise, add the 64-bit version `ASL.lib` to the project.



9. Make sure the build target of ASLClientCodeExample is **Executable** (**Build»Target Type**).
10. Build project ASLClientCodeExample.

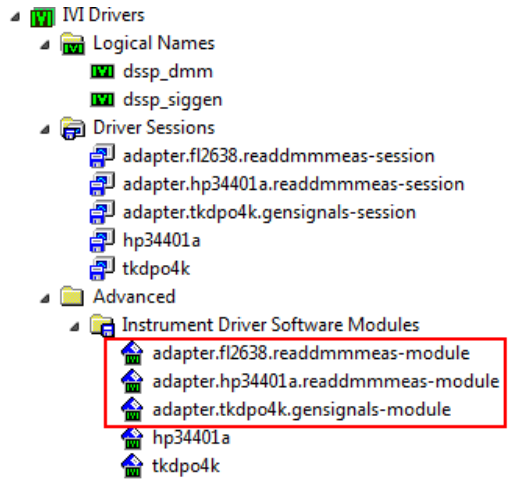
3.System Configuration

You can either configure your system manually in MAX, or you can use the pre-populated `IviConfigurationStore.xml` file that is provided in the zip file. Make sure to backup your existing `IviConfigurationStore.xml` file under `C:\ProgramData\IVI Foundation\IVI`, and then copy the provided `IviConfigurationStore.xml` file to the same location.

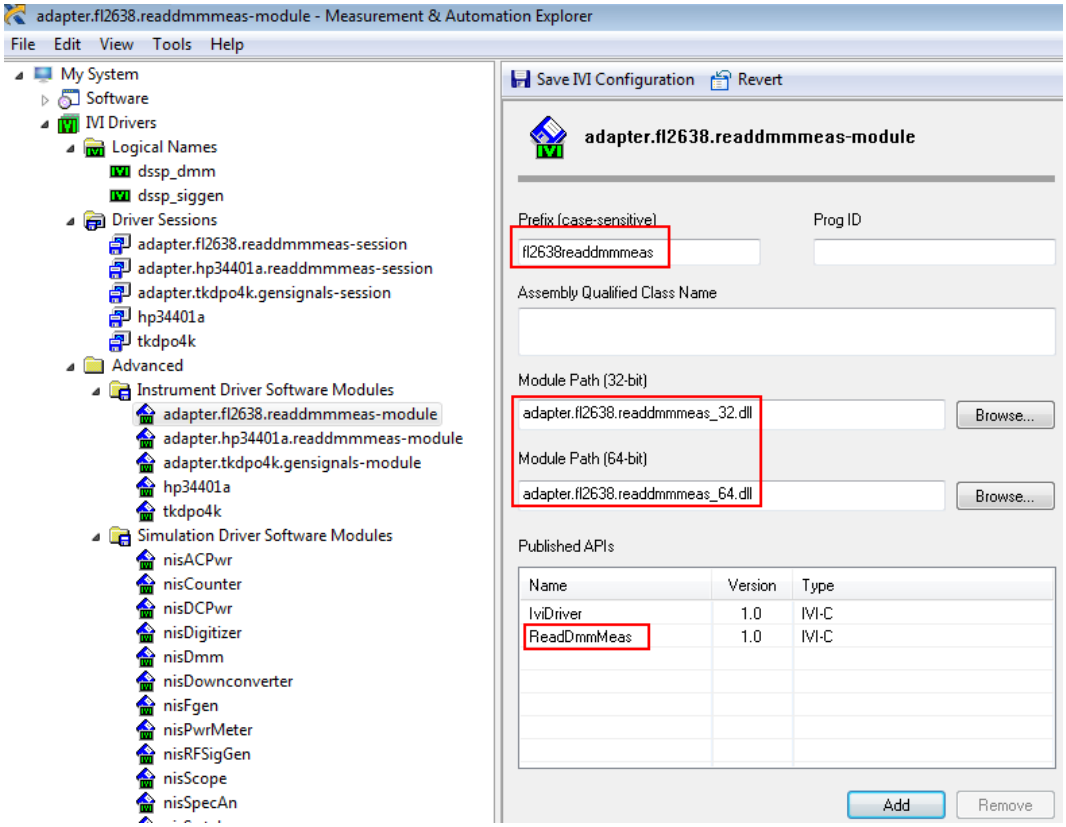
Complete the following steps to configure your system manually in MAX.

Note: Save each configuration before moving on to the next configuration step.

1. Create three software modules: adapter.fl2638.readdmmmeas-module, adapter.hp34401a.readdmmmeas-module, adapter.tkdpo4k.gensignals-module.



2. Configure the General page of each software module, as shown in the following image.

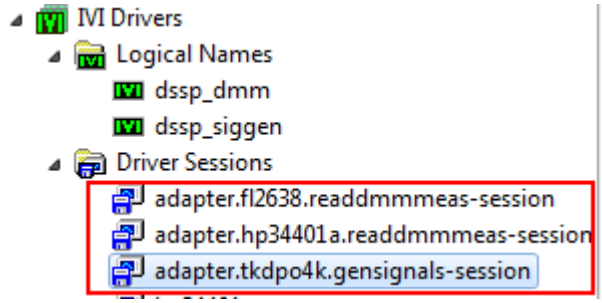


Note: The Published APIs will not pre-populate an option for ReadDmmMeas or GenSignals. You need to type it in manually.

The following table shows the settings for each software module:

Software Module	Prefix	Module Path (32-bit), change "_32" to "_64" for 64-bit	Published APIs
adapter.fl2638.readdmmmeas-module	fl2638readdmmmeas	`adapter.fl2638.readdmmmeas_32.dll`	ReadDmmMeas
adapter.hp34401a.readdmmmeas-module	hp34401areaddmmmeas	`adapter.hp34401a.readdmmmeas_32.dll`	ReadDmmMeas
adapter.tkdpo4k.gensignals-module	tkdpo4kgensignals	`adapter.tkdpo4k.gensignals_32.dll`	GenSignals

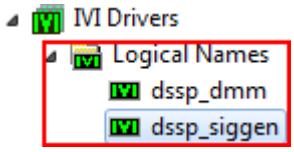
3. Create three driver sessions: adapter.hp34401a.readdmmmeas-session, adapter.fl2638.readdmmmeas-session and adapter.tkdpo4k.gensignals-session.



4. Configure each driver session. The following table shows the settings for each driver session.

Driver Session	Simulate With (General Tab)	Hardware (Hardware Tab)	Software Module
adapter.fl2638.readdmmmeas-session	Specific Driver	Add assets if you have live instrument	adapter.fl2638.readdmmmeas-module
adapter.hp34401a.readdmmmeas-session	Specific Driver	Add assets if you have live instrument	adapter.hp34401a.readdmmmeas-module
adapter.tkdpo4k.gensignals-session	Specific Driver	Add assets if you have live instrument	adapter.tkdpo4k.gensignals-module

5. Create two logical names: dssp_dmm and dssp_siggen.

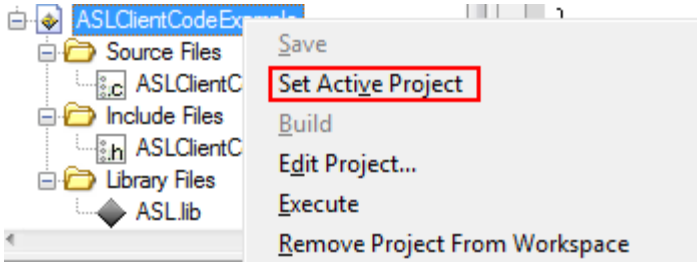


6. Link the two logical names with the proper driver session. Link dssp_dmm with adapter.fl2638.readdmmmeas-session or adapter.hp34401a.readdmmmeas-session. Link dssp_siggen with adapter.tkdpo4k.gensignals-session.

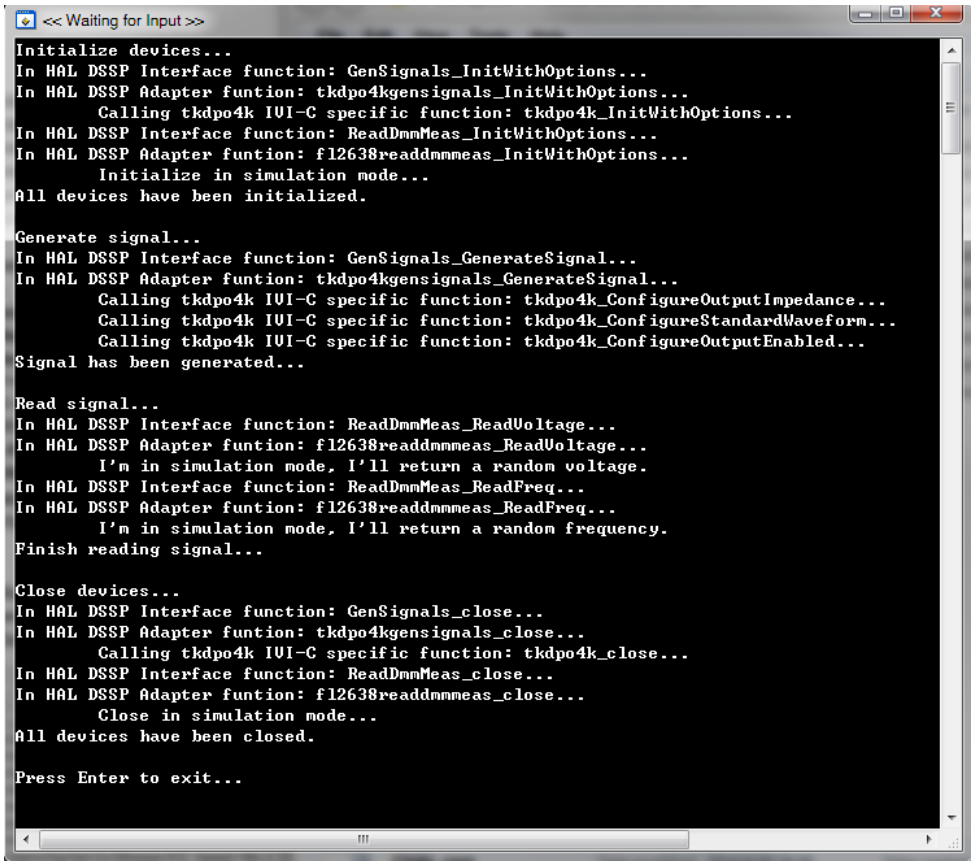
4. Run Example

After configuring the Logical Names, Driver Sessions and Software Modules, you can run the example.

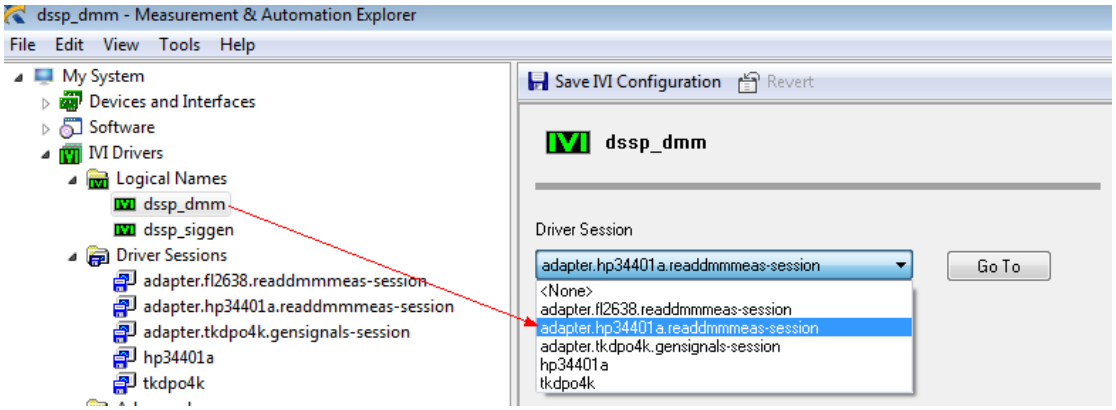
1. Set project ASLClientCodeExample as active project, then run it.



The following image displays the calling sequence. In MAX, dssp_dmm points to adapter.fl2638.readdmmmeas-session, dssp_siggen points to adapter.tkdpo4k.gensignals-session.



You can make dssp_dmm point to adapter.hp34401a.readdmmmeas-session if you want to use the Keysight 34401A to do the measurements.



You do not need to change any of your test code. Running the example again displays the calling sequence, as shown in the following image.