



IVI Instrument Driver Programming Guide (VEE Edition)

July 2012 Revision 2.0

1- Overview

1-1 Recommendation Of IVI-COM Driver

Agilent VEE added IVI-COM support since Ver7. This guidebook recommends using IVI-COM instrument drivers.

Notes:

- This guidebook shows examples that use KikusuiPwx IVI instrument driver (KIKUSUI Pwx series DC Power Supply). You can also use IVI drivers for other vendors and other models in the same manner.
- This guidebook describes how to use IVI-COM drivers with VEE Pro 9.3.

2- Creating Application

After launching VEE Pro, an empty workspace titled as "Main" is created. (When creating a new empty workspace, choose **File | New** menu.)

2-1 Adding Instrument

What you should do first is add an instrument that you want to use to the VEE Pro development environment. See the **Instrument Manager** pane that is located in upper-right of the screen. If it is not show, choose **I/O | Instrument Manager** menu to show.

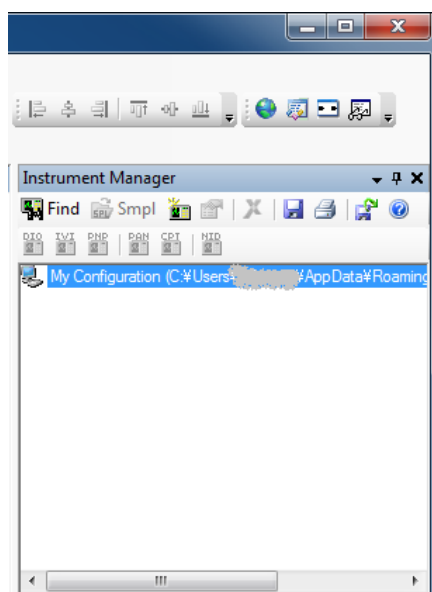


Figure 2-1 Instrument Manager

There are couple ways to add instruments and slightly different depending wheatear the instruments are already searched or not. This example assumes that an instrument (Kikusui PWX series DC supply) having IP address 192.168.1.5 connected with LAN interface.

Notes:

- When controlling instruments in VEE environment, it is recommended to search the instruments in advance using Connection Expert utility bundled with Agilent IO Libraries Suite. This is because VEE's instrument search function is cooperating with Agilent IO Libraries.

When Already Searched by Agilent Connection Expert

If the target instrument is already searched by Connection Expert, right-click on the Instrument Manager pane and then select **Find Instrument...** menu to bring up the **Identify Instrument** dialogue.

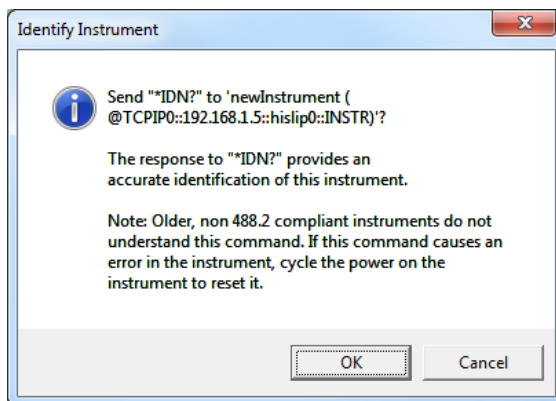


Figure 2-2 Identify Instrument dialogue

Verify the VISA address shown there then click OK. Continuously it ask you if sending an *IDN? query so click OK again..

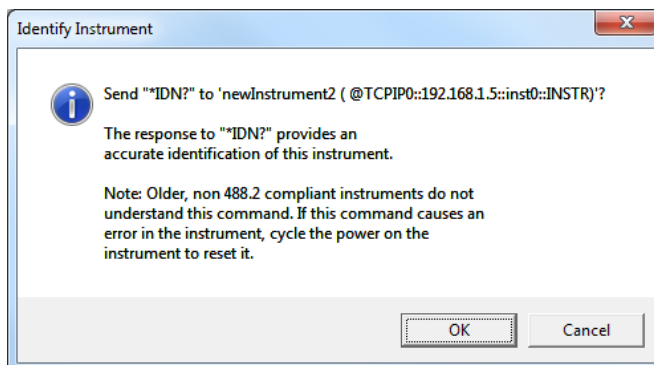


Figure 2-3 Identify Instrument dialogue

As completing model ID inquiry, the instrument list on Instrument Manager will be updated. As for PWX series DC power supply, the three LAN instrumentation protocols - VXI-11, HiSLIP, SCPI-RAW are supported, and VXI-11 and HiSLIP VISA addresses will be shown in VEE.

Select an instrument (HiSLIP VISA address in this example) on Instrument Manager, right-click on it and select **Instrument Properties...** menu.

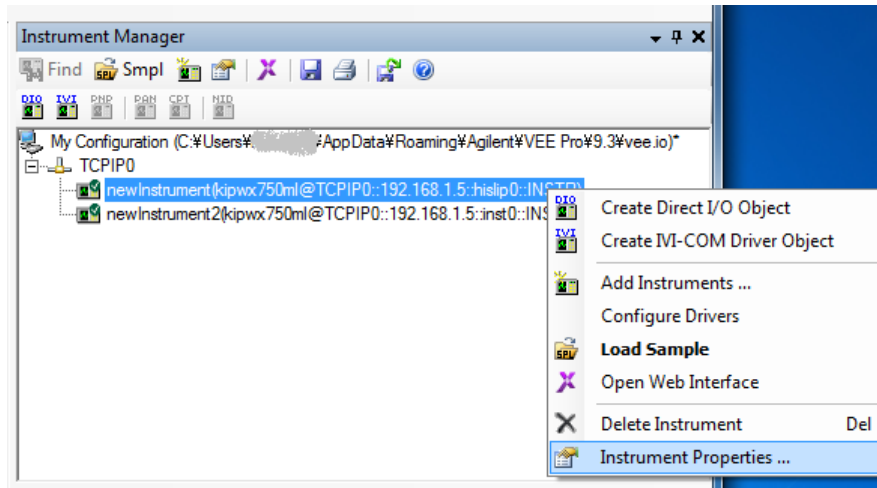


Figure 2-4 Instrument Manager

Then the following **Instrument Properties** dialogue will appear.

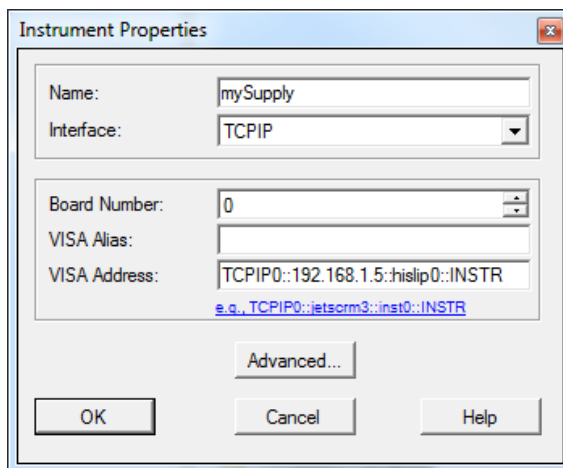


Figure 2-5 Instrument Properties

Here, type **mySupply** at **Name** field. Leave **VISA address** as is because it must be already correct one. Then click the **Advanced...** button. The **Advanced Instrument Property** dialogue will appear then select **IVI-COM Driver** tab.

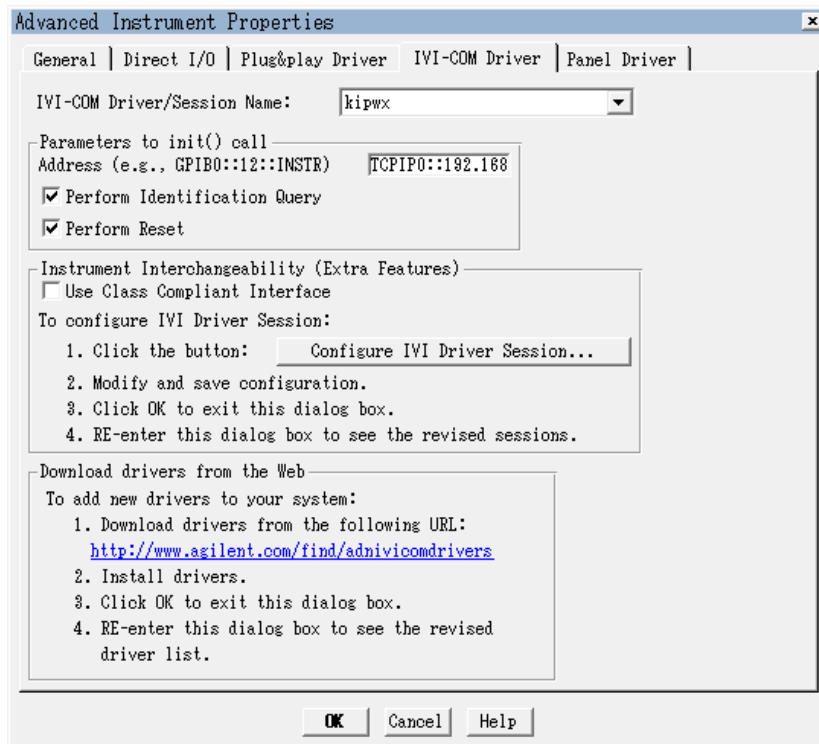


Figure 2-6 Advanced Instrument Properties

If the target instrument is already searched by Connection Expert, the correct IVI-COM instrument driver name is already shown in **IVI-COM Driver/Session Name**. For the case of KikusuiPwx instrument driver the correct setting is kipwx. (When not selected correctly, select **kipwx**.) After completing confirmation click OK to close the dialogue. You will be back to Instrument Properties then again click OK to close it.

Now the instrument name on Instrument Manager is changed to **mySupply**.

When Not Searched By Agilent Connection Expert

In this case, right-click on Instrument Manager and select **Add Instrument...** Then **Add Interface/Device** dialogue will appear.

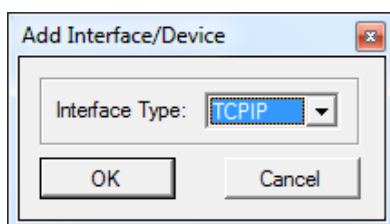


Figure 2-7 Add Interface/Device

GPIB is selected as default, but you control the instrument connected through LAN so select TCPIP and click OK. Now above-mentioned **Instrument Properties** dialogue will appear and the hereafter the procedure is basically similar. However, the instrument is not searched yet so you need type TCPIP::192.168.1.5::hislip0::INSTR at **VISA Address** field. Plus, IVI-COM instrument driver is not properly set in **Advanced Instrument Properties** yet, so you need confirm this by clicking the **Advanced...** button.

2-2 Creating IVI-COM Driver Object

Right-click on Instrument Manager then select **Create IVI-COM Driver Object** menu. Click anywhere you like on the workspace that is titled as Main to show a script window that controls your instrument.

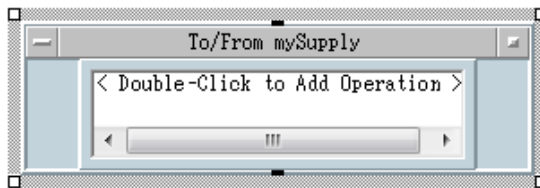


Figure 2-8 Script Window

Now double-click the line where **<Double-Click to Add Operation>** is written. then the **Select an Operation** dialogue will appear.

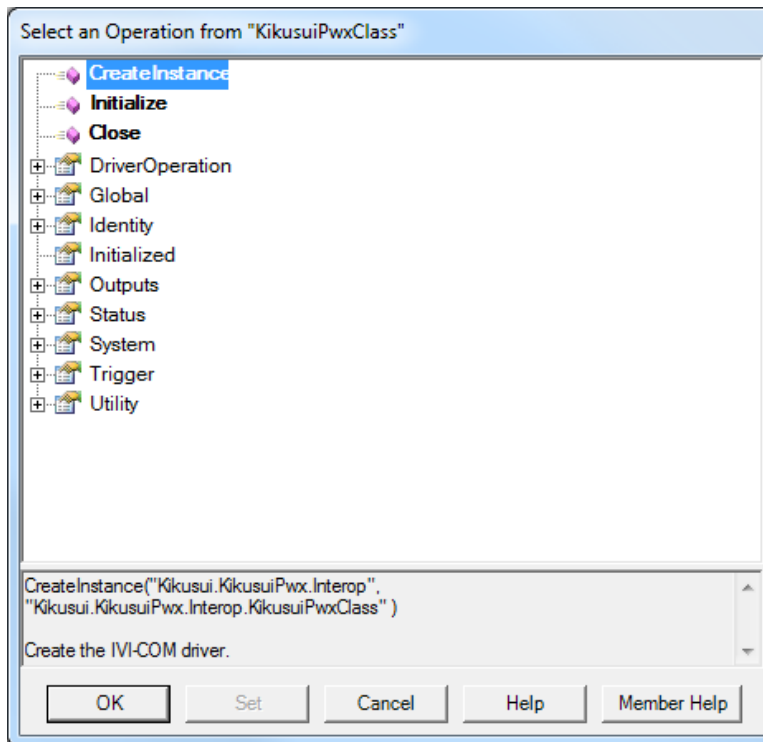


Figure 2-9 Select an Operation

Here it is required to create an instance of the driver object, so select **CreateInstance** and click OK. Then the screen switches to **Edit "CreateInstance"** dialogue.

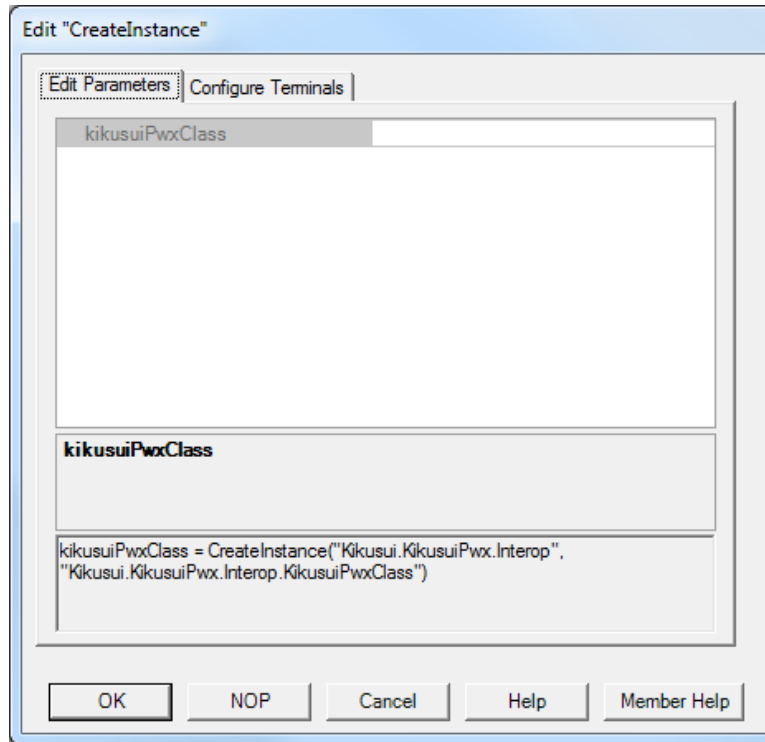


Figure 2-10 Edit "CreateInstance"

Because there is nothing to specify for **CreateInstance**, just click OK.

Now a code that creates the object has been added..

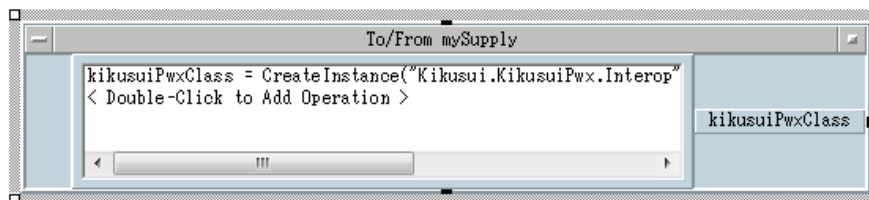


Figure 2-11 Script Window

2-3 Opening IVI Driver Session

Double-click the **<Double-Click to Add Operation>** part. Then the **Select an Operation** dialogue will appear and this time select **Initialize**.

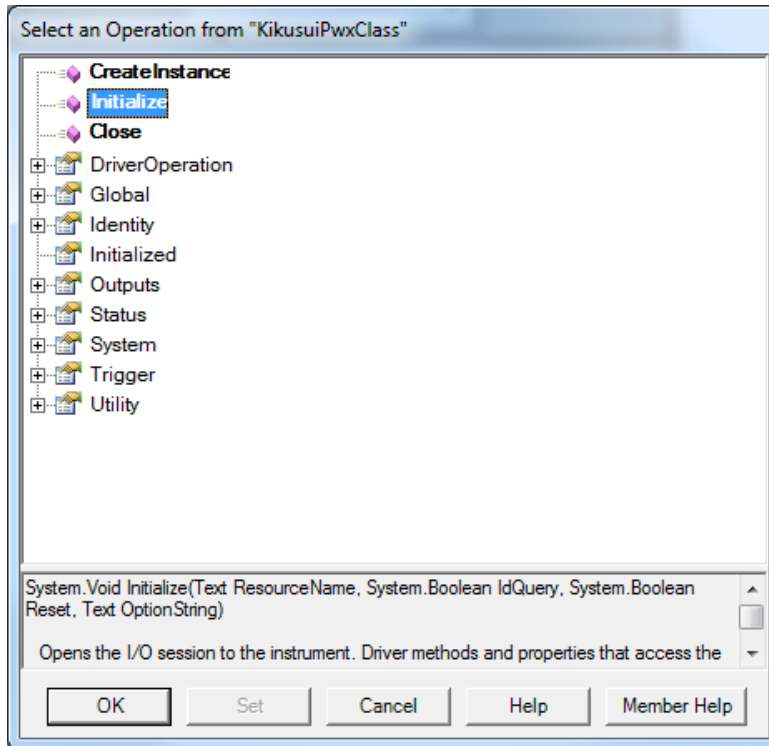


Figure 2-12 Select an Operation

Furthermore click OK to switch to **Edit "Initialize"** dialogue.

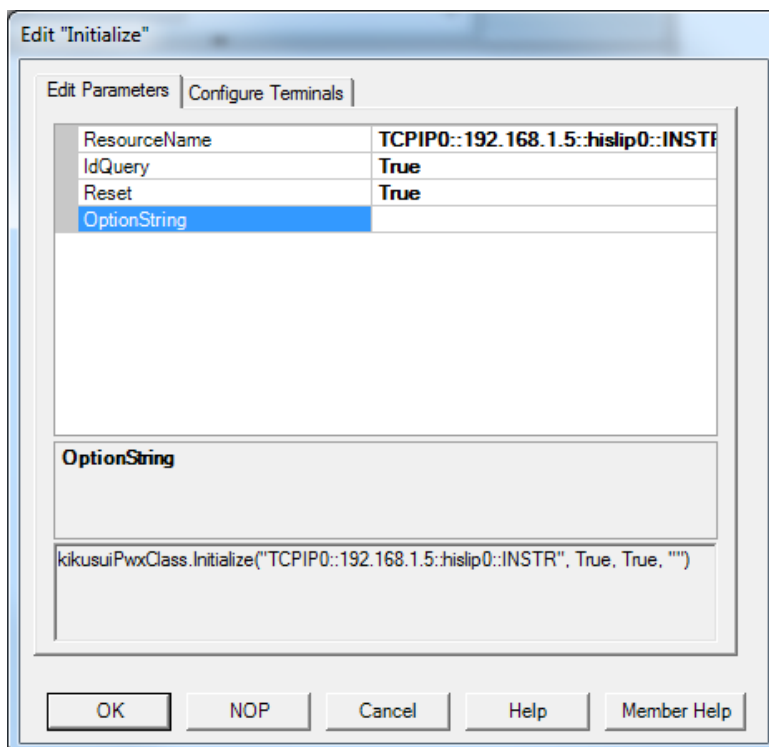


Figure 2-13 Edit "Initialize"

At **ResourceName** the correct VISA address must be already set. Keep **IdQuery** and **Reset** default value - True. At **OptionString** type **QueryInstrStatus=TRUE**.

Now let's talk about the parameters for the **Initialize** method. Every IVI-COM instrument driver has an **Initialize** method that is defined in the IVI specifications. This method has the following parameters.

Table 2-1 Params of Initialize Method

Parameter	Type	Description
ResourceName	string	VISA resource name string. This is decided according to the I/O interface and/or address through which the instrument is connected. For example, a LAN-based instrument having IP address 192.168.1.5 will be TCPIP::192.168.1.5::INSTR (when VXI-11 case).
IdQuery	Boolean	Specifying TRUE performs ID query to the instrument.
Reset	Boolean	Specifying TRUE resets the instrument settings.
OptionString	string	Overrides the following settings instead of default: RangeCheck Cache Simulate QueryInstrStatus RecordCoercions Interchange Check Furthermore you can specify driver-specific options if the driver supports DriverSetup features.

ResourceName specifies a VISA resource. If **IdQuery** is TRUE, the driver queries the instrument identities using a query command such as "***IDN?**". If **Reset** is TRUE, the driver resets the instrument settings using a reset command such as "***RST**".

OptionString has two features. One is what configures IVI-defined behaviours such as **RangeCheck**, **Cache**, **Simulate**, **QueryInstrStatus**, **RecordCoercions**, and **Interchange Check**. Another one is what specifies **DriverSetup** that may be differently defined by each of instrument drivers. Because the **OptionString** is a string parameter, these settings must be written as like the following example:

```
QueryInstrStatus = TRUE , Cache = TRUE , DriverSetup=12345
(DriverSetup=12345 is only an imaginary parameter for explanation. )
```

Names and setting values for the features being set are case-insensitive. Since the setting values are Boolean type, you can use any of TRUE, FALSE, 1, and 0. Use commas for splitting multiple items. If an item is not explicitly specified in the **OptionString** parameter, the IVI-defined default value is applied for the item. The IVI-defined default values are TRUE for **RangeCheck** and **Cache**, and FALSE for others.

Some instrument drivers may have special meanings for the **DriverSetup** parameter. It can specify items that are not defined by the IVI specifications when invoking the **Initialize** method, and its purpose and syntax are driver-specific. Therefore, specifying the **DriverSetup** must be at the last part on the **OptionString** parameter. Because the contents of **DriverSetup** are different depending on each driver, refer to driver's Readme document or online help.

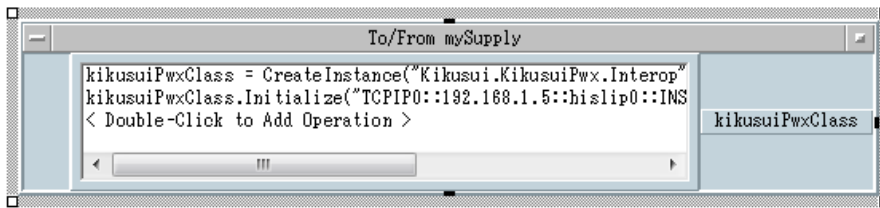


Figure 2-14 Script Window

After adding operations the lowest line still shows **<Double-Click to Add Operation>**, so you can add more operations in the same manner.

2-4 Setting Voltage And Current

Here, you add 20V, 2A, and OUTPUT ON settings. This adds each of **VoltageLevel**, **CurrentLimit**, and **Enabled** properties as an operation.

Voltage Setting

The operation you add here is **Outputs->Item(Name)->VoltageLevel** in the tree hierarchy. **VoltageLevel** is a property. The buttons in the **Select an Operation** dialogue will show **Get/Set** choices. Here click **Set** because you "set" a value.

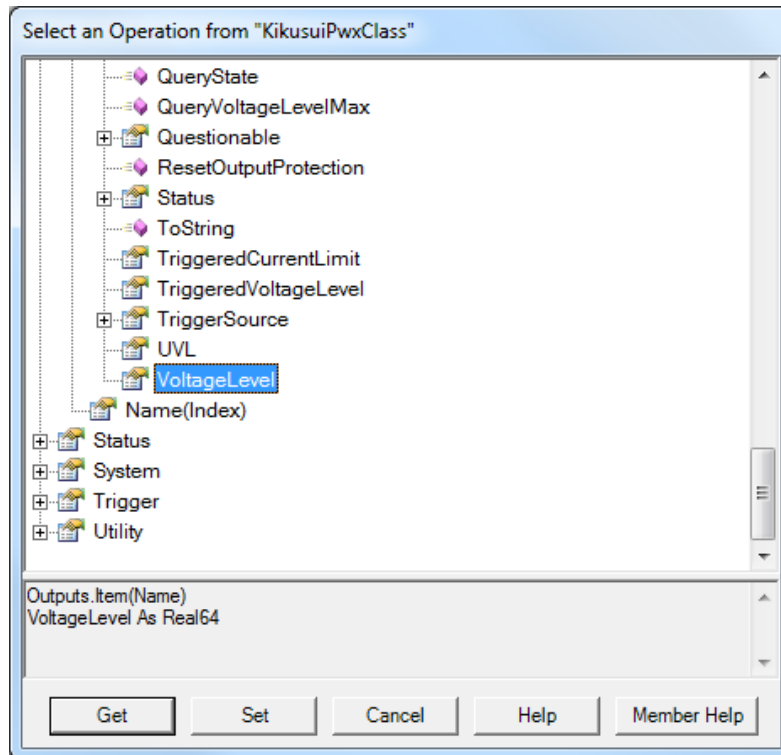


Figure 2-15 Select an Operation

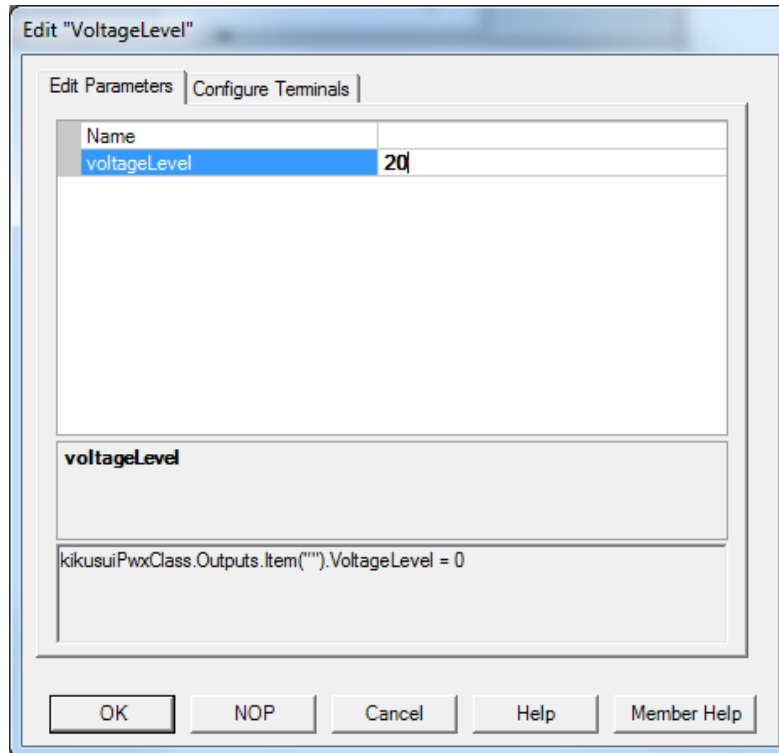


Figure 2-16 Edit "VoltageLevel"

Set **20** in the **Edit "VoltageLevel"** dialogue.

Current Setting

The operation you add here is **Outputs->Item(Name)->CurrentLimit** in the tree hierarchy. This is also a property, so select **Set** from the lowest buttons in **Select an Operation** dialogue. Specify **2** as a setting value.

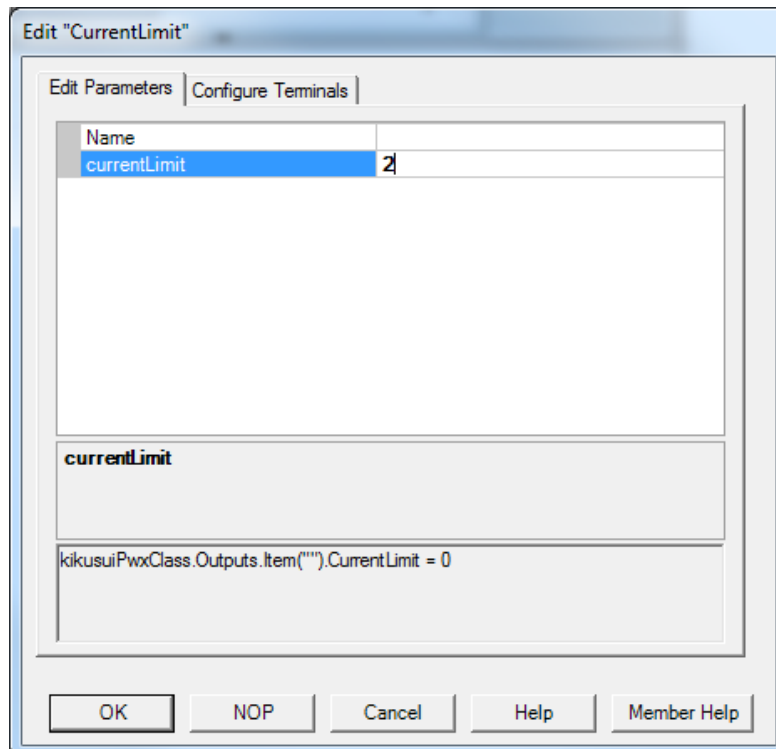


Figure 2-17 Edit "CurrentLimit"

Output Setting

The operation you add here is **Outputs->Item(Name)->Enabled** in the tree hierarchy. This is also a property, so select **Set** from the lowest buttons in **Select an Operation** dialogue. Specify **True** as a setting value.

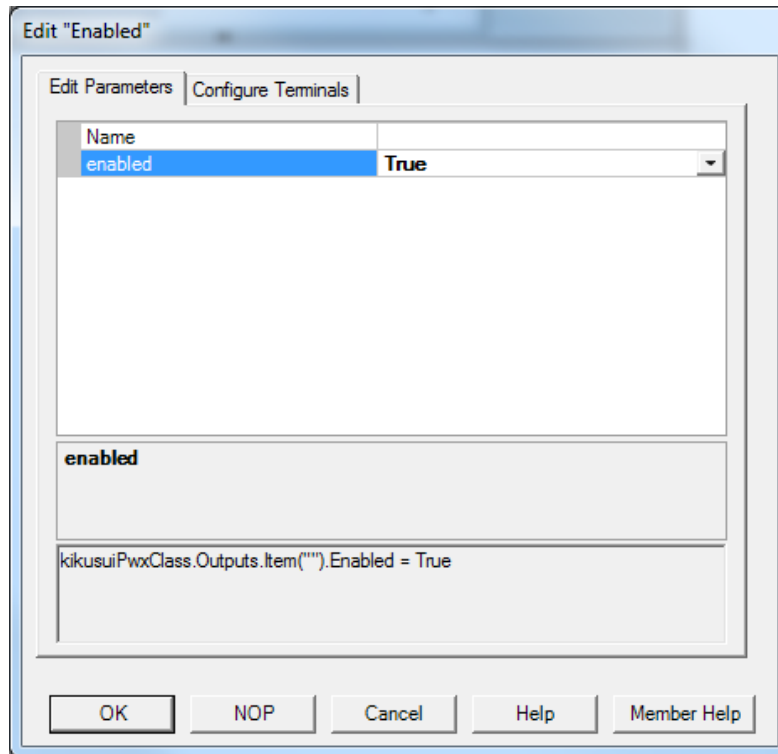


Figure 2-18 Edit "Enabled"

2-5 Closing IVI Driver Session

Finally close the driver session. Here select **Close**.

2-6 Execution

At the conditions so far, , the instrument control program will be like below.

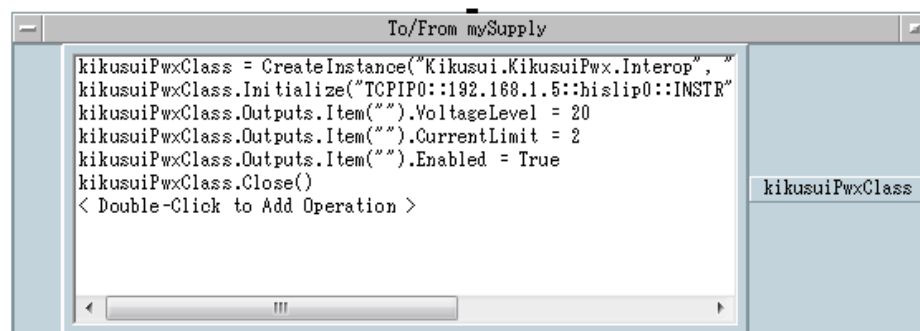


Figure 2-19 Script Window

Execute the program by selecting **Debug | Run/Resume** menu. If there is no programming fault and the instrument is properly connected, , Voltage 20V, Current 2A, and Output ON settings will be made.

IVI Instrument Driver Programming Guide

Product names and company names that appear in this guidebook are trademarks or registered trademarks of their respective companies.

©2012 Kikusui Electronics Corp. All Rights Reserved.