

## Using Mini-Circuits' Portable Test Equipment (PTE) Software - Frequently Asked Questions (FAQs)

### Keyword Cross Reference

.NET.....	3, 4, 6, 7, 8	Linux .....	8
32-bit .....	3, 4, 5, 6, 8, 17	MatLab .....	4, 9
64-bit .....	3, 4, 5, 6, 8, 17	Network.....	3, 11, 16
ActiveX.....	3, 4, 5, 8, 10, 12, 13	Perl.....	4
Attenuator .....	14, 15, 16, 18, 19	Power Sensor .....	10, 17
CVI.....	9	Python.....	4, 8, 9
DLL .....	3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16, 17, 18	RS232 .....	15, 17, 18
Error.....	12, 13	RS232/USB-SPI converter .....	17, 18
Ethernet.....	3, 8, 9, 11, 14, 16	SPI .....	17, 18
Frequency Counter .....	19	Telnet.....	3, 8, 14, 16
GUI.....	3, 15, 17	USB .....	3, 4, 5, 6, 8, 12, 13, 14, 16, 17, 18
HTTP.....	3, 8, 9, 11, 14, 16	VEE.....	9
IP Address .....	3, 11, 16	Visual Studio .....	4
LabVIEW.....	4, 10, 11, 12, 13, 14	Windows.....	4, 5, 6, 7, 8, 12, 16, 17
LabWindows .....	9		

### Questions

1. What control options are supported by Mini-Circuits' PTE Series? .....	3
2. How do I configure the Ethernet settings (IP address, gateway, etc) for my PTE device? .....	3
3. Can I create 32-bit and 64-bit applications using the DLL? .....	3
4. Should I use the ActiveX or .NET DLL to Control Mini-Circuits PTE via USB? .....	4
5. Where do I install the ActiveX DLL file for USB control? .....	5
6. Where do I install the .NET DLL for USB control? .....	6
7. Where can I get the header files for the DLLs? .....	6
8. I've installed the DLL but the programming environment can't access it.....	7
9. Can your test equipment run under a Linux operating system? .....	8
10. Can I use Python 64-bit for USB control with the DLLs? .....	8
11. How does Python interpret DLL functions that take arguments by value? .....	9
12. Can I use National Instruments LabWindows / CVI, Keysight VEE® or MatLab®? .....	9
13. Do you provide LabVIEW VIs for your test equipment? .....	9
14. How do I set the frequency compensation factor for power sensor measurements in LabVIEW?....	10
15. How can I control Mini-Circuits PTE over a network using LabVIEW? .....	11
16. How do I resolve error number -2147221164 when running my LabVIEW VI? .....	12
17. How do I resolve error number -3005 when selecting the DLL object in LabVIEW? .....	12
18. How do I resolve error number -2147221231 when running my LabVIEW VI? .....	13
19. Why can't I get the "set SP4T switch" command to work via Ethernet? .....	14

20. How quickly can I sweep the attenuation on a RUDAT / RCDAT programmable attenuator? .....	14
21. What attenuation level will the RUDAT / RCDAT programmable attenuators power up with?.....	15
22. Can I set the attenuation of a RUDAT attenuator and then disconnect the computer? .....	15
23. Can I set the attenuation of a RCDAT attenuator and then disconnect the computer? .....	16
24. Can the RCDAT attenuator series be controlled without using a DLL?.....	16
25. How are power sensor offset values interpolated?.....	17
26. How can I control my PWR-SEN-6G(+) power sensor in Windows 7? .....	17
27. Can the PWR series power sensors interface with other test equipment by USB?.....	17
28. Which SPI modes are supported in the RS232/USB-SPI converter GUI? .....	17
29. What is the purpose of the LE connection on the RS232/USB-SPI converter? .....	18
30. How do I send more than 16 data bits using the RS232/USB-SPI converter? .....	18
31. When should I manually set the input range parameter of the frequency counter?.....	19

### 1. What control options are supported by Mini-Circuits' PTE Series?

Mini-Circuits' PTE products can be controlled programmatically from a range of programming environments via their USB or Ethernet connections. Control via Ethernet is achieved using a simple ASCII text based command set, sent using HTTP or Telnet. HTTP in particular is commonly supported in a range of environments, often without the need for any external libraries, making it a particularly convenient control method. Control integration via USB is accomplished using Mini-Circuits' ActiveX or .NET DLL files (where supported) or USB interrupt codes.

### 2. How do I configure the Ethernet settings (IP address, gateway, etc) for my PTE device?

All Mini-Circuits PTE devices are configured by default to use DHCP (Dynamic Host Control Protocol). This means that on many networks that support DHCP (typical of most small office and home networks), the device will automatically be assigned an IP address by the network when it is plugged in so can be used straight away with no manual configuration needed. For more complex networks where DHCP is not supported, or where the user wishes to configure specific IP settings, the device must first be connected by USB. The supplied Mini-Circuits GUI allows full manual Ethernet configuration (gateway, subnet mask, IP address, TCP/IP port for HTTP communication and password) to be stored in the device's memory through the USB connection. Following this configuration stage, the device can then be controlled over the network with the new settings and the USB connection need not be used again (unless a change to the Ethernet configuration is required later).

### 3. Can I create 32-bit and 64-bit applications using the DLL?

Mini-Circuits' ActiveX and .NET DLLs can all be used on both 32-bit and 64-bit operating systems. The matrix of program types that can be created is summarised below (a 64-bit programming environment is required to create a 64-bit application).

	Operating System Type	
	32-bit	64-bit
ActiveX DLL	Creation of 32-bit applications only	Creation of 32-bit applications only
.NET DLL	Creation of 32-bit applications only	Creation of 32-bit and 64-bit applications

**Table 1 - Summary of application types possible with the DLLs**

#### 4. Should I use the ActiveX or .NET DLL to Control Mini-Circuits PTE via USB?

The choice of which of the 2 Mini-Circuits DLL files to use (ActiveX or .NET) is usually a question of which of those frameworks is supported by the programming environment. A matrix summarising support for ActiveX and .NET between some common programming environments is included below. Where you have a free choice (ie: the environment supports ActiveX and .NET) we would typically recommend the .NET DLL since this does not need to be registered in the Windows operating system, it can just be placed in the project folder and referred to from within the program. The functionality exposed by the 2 DLLs is the same so either can be chosen from that point of view.

Programming Environment	Programming Environment Type	
	32-bit	64-bit
Visual Studio 6.0 (for C++, C#, VB)	ActiveX DLL supported	N/A
Visual Studio 2002 (.NET) or later (for C++, C#, VB.NET)	ActiveX and .NET DLLs supported	ActiveX and .NET DLLs supported
LabVIEW 8.0	ActiveX DLL supported	N/A
LabVIEW 2009 or later	ActiveX and .NET DLLs supported	.NET DLL supported
MATLAB 7 or later	ActiveX DLL supported	N/A
MATLAB 2009 or later	ActiveX and .NET DLLs supported	ActiveX and .NET DLLs supported
Python (CPython distribution)	ActiveX DLL supported	Neither DLL supported <sup>1,2</sup>
Perl	ActiveX DLL supported	Neither DLL supported <sup>1,2</sup>

**Table 2 - Summary of ActiveX and .NET support in some programming environments**

**Notes:**

1. Open source environments such as Perl and Python are available in multiple distributions. Some 64-bit distributions (eg: IronPython and Python.NET) do provide support for the .NET framework and may therefore be capable of working with Mini-Circuits' .NET DLLs, however this has not currently been verified by Mini-Circuits.
2. For 64-bit distributions that do not support .NET, Mini-Circuits can often supply an executable program that will interface with the Mini-Circuits DLL and PTE hardware. The executable can be called from the Python/Perl script to allow control over the Mini-Circuits PTE, without requiring the environment to support the .NET framework. Please contact Mini-Circuits' Applications department for support.

## 5. Where do I install the ActiveX DLL file for USB control?

1. The ActiveX DLL should be placed in the Windows system folder as below:
  - For 32-bit operating systems: `C:\Windows\System32\`
  - For 64-bit operating systems: `C:\Windows\SysWOW64\`
2. Run the Command Prompt in elevated mode by right-clicking on it and selecting "Run as administrator."

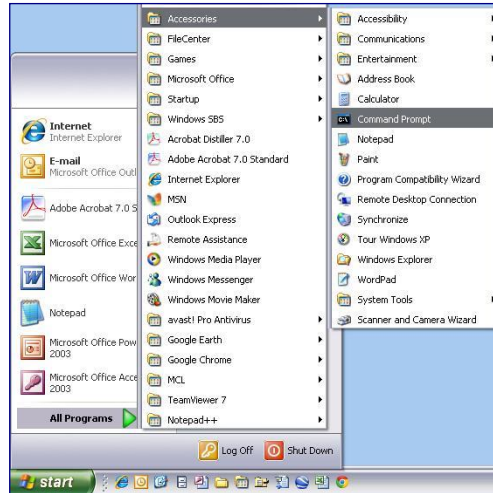


Figure 1 - Opening the Command Prompt in Windows XP

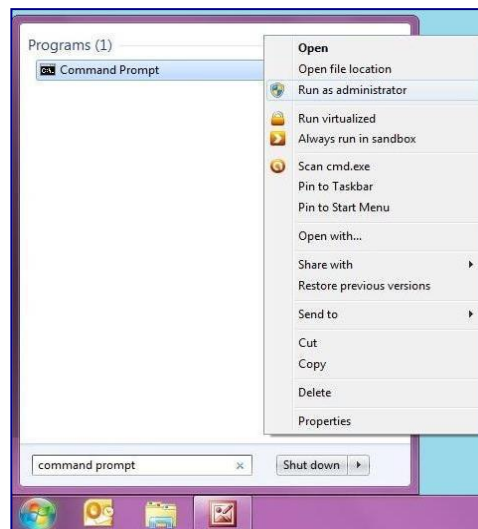


Figure 2 - Opening the Command Prompt in Windows 7

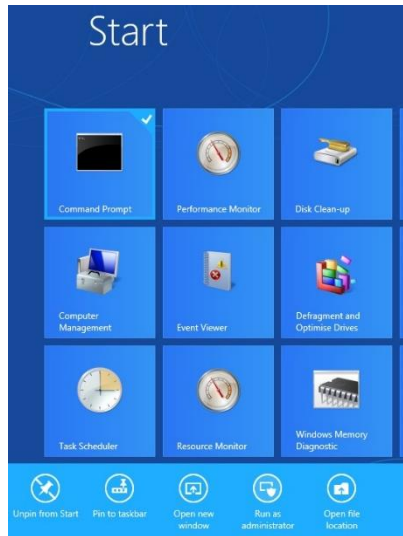


Figure 3 - Opening the Command Prompt in Windows 8

3. Use regsvr32 to register the DLL in the command prompt:
  - For 32-bit operating systems type `regsvr32 dll_filename.dll`
  - For 64-bit operating systems type:
    - a. `cd \windows\syswow64` (to move to the correct directory)
    - b. `regsvr32 dll_filename.dll` (to register the DLL)

## 6. Where do I install the .NET DLL for USB control?

The .NET DLL file just needs to be copied to the project directory that you're using, it does not need to be registered.

## 7. Where can I get the header files for the DLLs?

Mini-Circuits' DLL files are ActiveX object and .NET class libraries so no additional header files are required. Please see the programming manuals for instructions and examples on their use.

8. I've installed the DLL but the programming environment can't access it  
Windows sometimes blocks system files that are downloaded from the Internet, allowing them to be installed but not accessed. To check this, find the .NET DLL on your PC, right-click on it and select Properties.

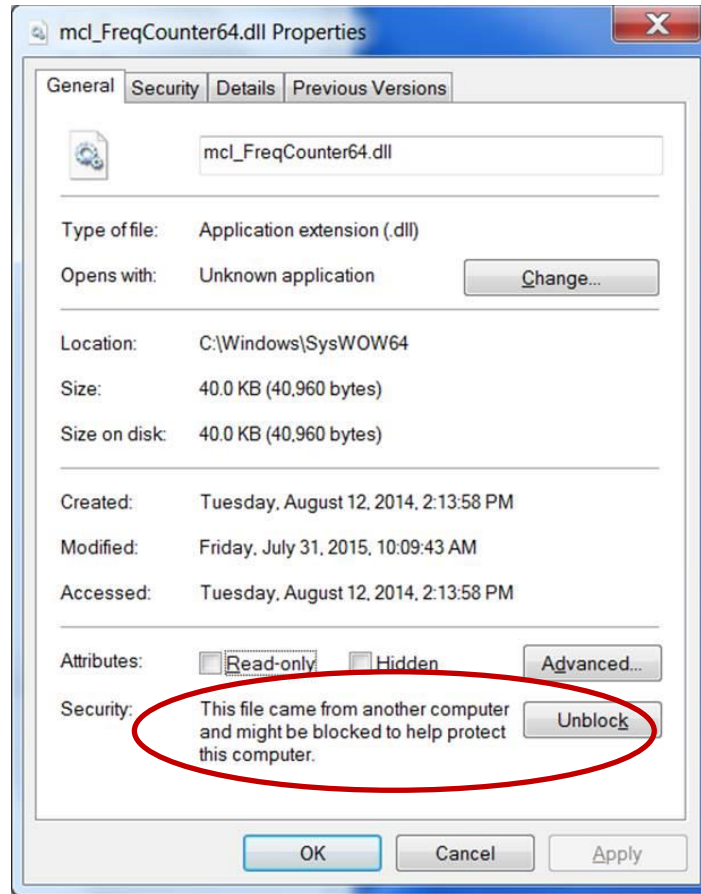


Figure 4 - DLL Properties dialog indicating the file has been blocked

If the security warning appears at the bottom, stating "This file came from another computer and might be blocked to help protect the computer." click the Unblock button to allow the DLL to operate.

## 9. Can your test equipment run under a Linux operating system?

Yes, all models support communication using USB interrupt codes which can be used in a Linux environment. Additionally, many models also have an Ethernet connection, allowing communication via HTTP or Telnet. Full details are in the relevant programming manuals.

## 10. Can I use Python 64-bit for USB control with the DLLs?

The standard 64-bit Python distribution does not support ActiveX or .Net DLLs. The 32-bit version of Python will work with our ActiveX DLL and is compatible with 64-bit versions of Windows 7. In general, unless you are working with very large memory allocations (>3GB) there is no advantage to the 64-bit Python over 32-bit, and the 32-bit version has wider compatibility with third party software.

There are 2 implementations of Python that do support .Net and those are IronPython and Python.NET. In these implementations you should be able to use our .NET DLL, with the following code:

```
>> import clr
>> clr.AddReference('mcl_pm64.dll')
>> from mcl_pm64 import usb_pm
>> my_instance = usb_pm()
```

If you're not able to use the above Python versions then the other option we have is to develop an executable that operates between the Python script and the DLL. This would be a simple program (written and compiled in C#) that references the DLL, uses it to send and read a command, disconnects the DLL, then terminates. To use this in your Python program you would just need to include a single line to call the program and pass some command line arguments (the serial number and the command that you want to send); you would be able to read the device's response in return. This approach means that the Python environment doesn't need to know anything about the DLL as this is wrapped up inside the executable.

Alternatively, if your PTE device has an RJ45 connection and you can control over Ethernet rather than USB, you can use HTTP for communication. This is a much simpler process since HTTP support is quite common in Python without the need for any of the above libraries/executables.



## 11. How does Python interpret DLL functions that take arguments by value?

Some DLL functions return a value (usually integer 0 or 1 to indicate success/failure) but take pointers to variables as arguments which are also updated during execution of the function to provide information on the device.

An example is `GetOffsetValues(ByRef short NoOfPoints,ByRef double FreqArray[],ByRef float LossArray[])` in the power meter DLL which returns 1 for success but provides the significant information (the purpose of the function) in the arguments. In this case, `NoOfPoints` provides the number of offset points set, `FreqArray` is an array of size `NoOfPoints` containing the frequency values of the offsets, and `LossArray` is an array of the corresponding dB offset values for those frequency points.

Since Python does not have a concept of "pointers" the variables cannot be accessed in this way. Instead, the return value of the function is a tuple which contains all that data, in the form:  
(Function\_Return\_Value, NoOfPoints, (FreqArray[0], FreqArray[1], ...FreqArray[NoOfPoints - 1]), (LossArray[0], LossArray[1], ...LossArray[NoOfPoints - 1]))

## 12. Can I use National Instruments LabWindows / CVI, Keysight VEE® or MatLab®?

Yes. The ActiveX® and .NET objects included in the software package allow these products to be used via the USB connection with any program that supports ActiveX® or .NET. In addition, these programming environments also support HTTP communication which allows Mini-Circuits' Ethernet enabled test products to be controlled over a network.

## 13. Do you provide LabVIEW VIs for your test equipment?

Example VIs are provided as part of the standard software package for all models, available for download from [http://www.minicircuits.com/support/software\\_download.html](http://www.minicircuits.com/support/software_download.html).

## 14. How do I set the frequency compensation factor for power sensor measurements in LabVIEW?

The majority of Mini-Circuit's PTE functions are achieved through "Invoke Nodes" when using the ActiveX DLL in LabVIEW. Since the compensation frequency of the power sensor is defined as a global property in the DLL, rather than a function, an "Invoke Property" block is needed to set it.

1. Select a new Property Node from the Connectivity >> ActiveX palette

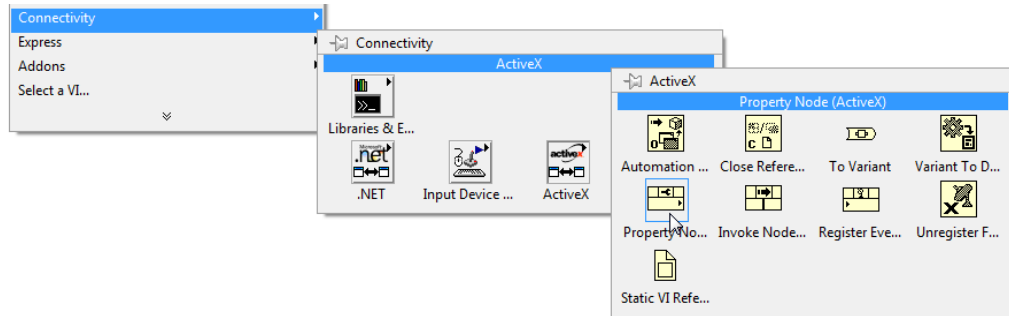


Figure 5 - Creating an Invoke Property

2. Place the property node on the block diagram, right-click on it and go to "Select Property" > "Freq"
3. LabView defaults the Property Node to "read" so right-click it a second time and select "Change To Write" to allow the property to be edited
4. Create a constant at the input of the Invoke Property to hold the frequency value to be set

## 15. How can I control Mini-Circuits PTE over a network using LabVIEW?

Mini-Circuits' PTE products support HTTP Get/Post commands for communicating over an Ethernet network. LabVIEW has built in support for this with an HTTP Get VI in the Data Communication >> Protocols >> HTTP Client palette. Place the HTTP Get object on the block diagram and pass it a string constant made up of the PTE device's IP address and the command to send, for example 192.168.9.67/:MN? to return the model name of a device with IP address 192.168.9.67. An indicator can be added the HTTP Get block to output the response from the PTE device.

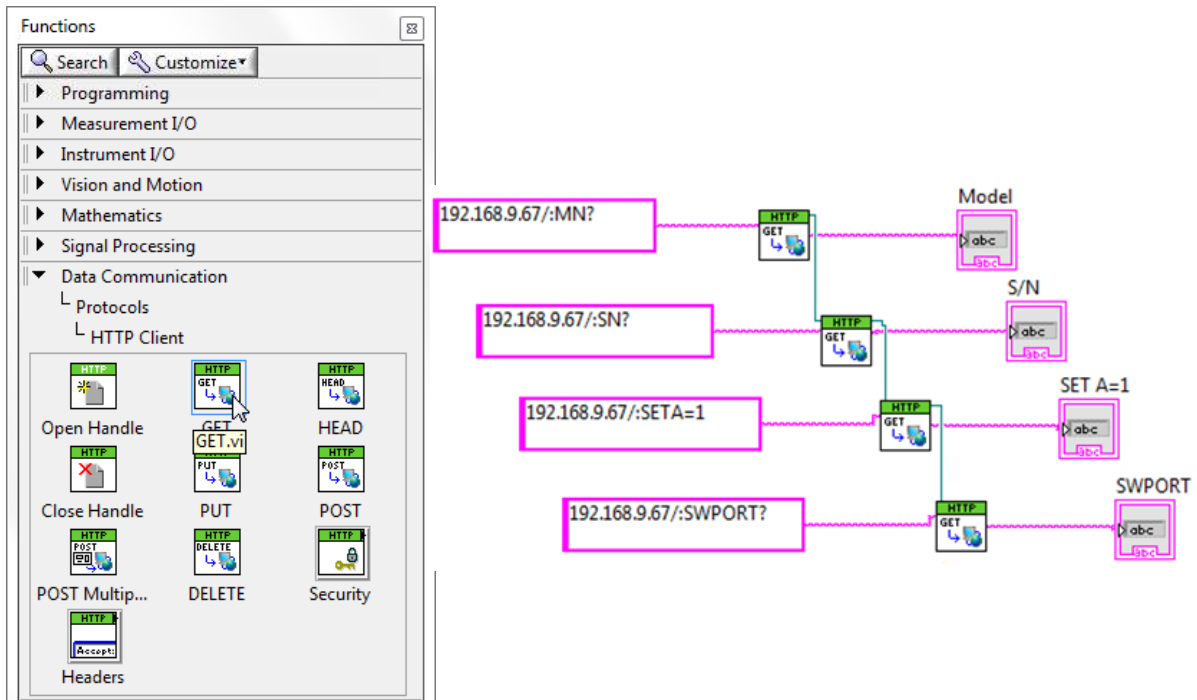


Figure 6 - LabVIEW's HTTP Client palette and an example of communication with Mini-Circuits' switches:

- 192.168.9.67/:MN? (return the model name for the switch box with IP address 192.168.9.67)
- 192.168.9.67/:SN? (return the serial number for the switch box with IP address 192.168.9.67)
- 192.168.9.67/:SETA=1 (set switch A to state 1 in the switch box with IP address 192.168.9.67)
- 192.168.9.67/:SWPORT? (return the switch state of the switch box with IP address 192.168.9.67)

## 16. How do I resolve error number -2147221164 when running my LabVIEW VI?

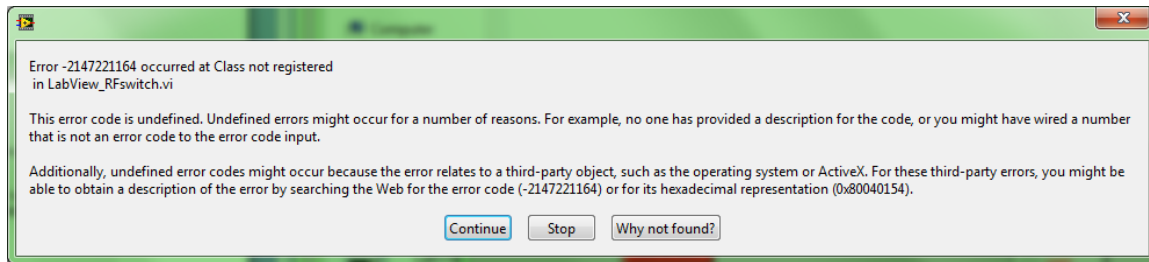


Figure 7 - Error message for error number -2147221164

This error occurs when the ActiveX DLL was used but not registered with the Windows operating system. Instructions on how to register your ActiveX DLL can be found in the relevant product programming manual on the Mini-Circuits website:

[http://www.minicircuits.com/support/software\\_download.html](http://www.minicircuits.com/support/software_download.html)

## 17. How do I resolve error number -3005 when selecting the DLL object in LabVIEW?

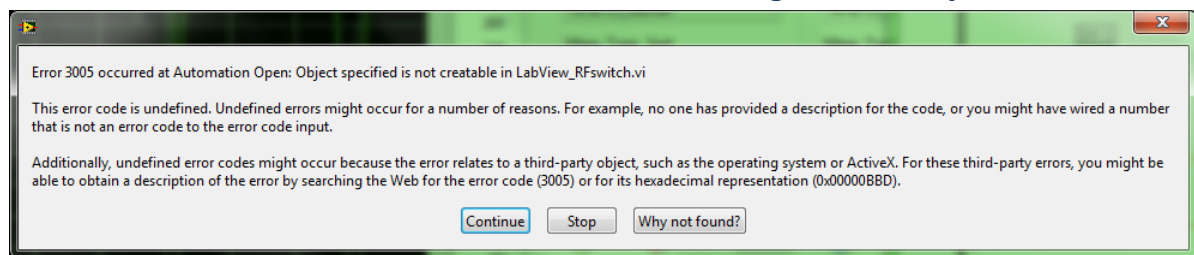


Figure 8 - Error message for error number -3005

This error occurs when setting up the Automation Refnum node on the block diagram and attempting to select an object type from the DLL which is not creatable. To solve this issue, check the the "Show Creatable Objects Only" options in the "Select Object from Type Library" dialog to remove the non-creatable objects; look for the USB control object that relates to your Mini-Circuits test product.

## 18. How do I resolve error number -2147221231 when running my LabVIEW VI?

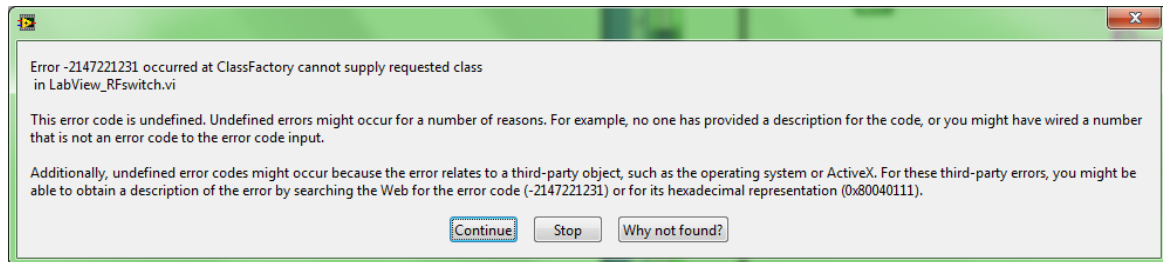


Figure 9 - Error message for error number -2147221231

The most common cause of this error encountered with Mini-Circuits' test systems is an issue with the program locating the correct version of the DLL. This can be an issue if the LabVIEW VI was created with a newer version of the DLL than is installed on the PC that needs to run the VI. The steps to solve this problem are as follows:

- a. Open the block diagram in LabVIEW (accessible by pressing Ctrl+E).
- b. Find the "Automation Refnum" control on the block diagram, right click it and go to the "Select ActiveX Class" option.
- c. Select "Browse..." and locate the correct DLL.

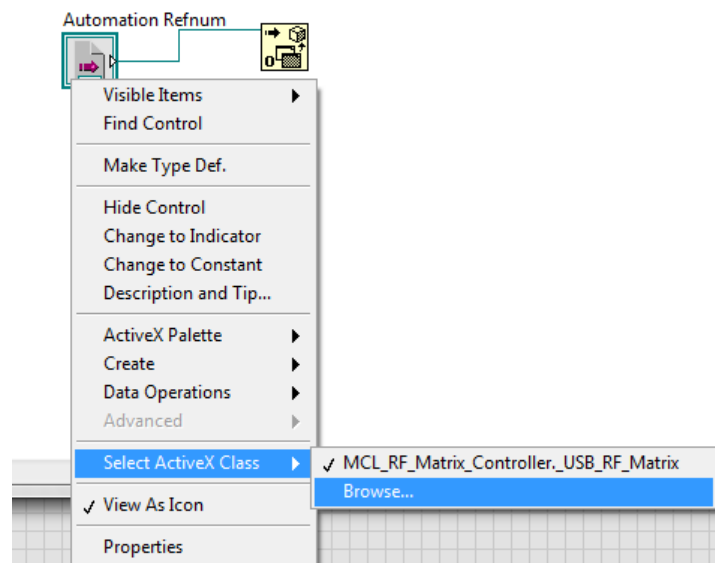


Figure 10 - Browsing for the correct DLL to use in the Automation Refnum object

- d. Check the "Show Creatable Objects Only" option when the "Select Object from Type Library" dialog opens to leave only 1 or 2 options (depending on the product).
- e. Select the USB control object relating to the Mini-Circuits' test product that you are trying to control.

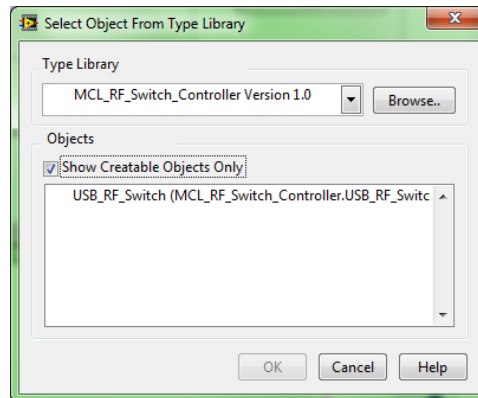


Figure 11: The Select Object from Type Library dialog with "Show creatable objects only" checked

- f. This should have refreshed the connection to the DLL but it may now be necessary to re-select each of the DLL methods in the LabVIEW Invoke Node blocks.

## 19. Why can't I get the "set SP4T switch" command to work via Ethernet?

I can connect to my RC-2SP4T-A18 (dual SP4T) via Telnet and HTTP and it responds to certain commands such as :MN? to return model name but the set and get SP4T switch state commands don't work.

All Mini-Circuits' switch boxes support the "Set Port" command ("SETP=x") where an integer value can be passed to set every switch in the switch box, regardless of whether it is an SPDT or SP4T switch. In addition, the latest firmware (from C8 onwards) also includes specific "Set SP4T" commands to in the form "SP4Tx:STATE:y" (ie: set switch number x to state y). If you have an older unit that does not support this more intuitive command set you can contact [testsolutions@minicircuits.com](mailto:testsolutions@minicircuits.com) with the details of your unit to obtain the latest firmware at no extra cost.

## 20. How quickly can I sweep the attenuation on a RUDAT / RCDAT programmable attenuator?

Mini-Circuits' RUDAT / RCDAT models include advanced firmware that allows you to program a fast sweep or hop sequence into the device's internal memory. This allows you to avoid the inherent communication delays (typically in the order of 2-10 ms or more over USB or Ethernet, depending on the PC configuration) when the software is issuing each individual attenuation command to the attenuator. This firmware allows a minimum dwell time in the order of hundreds of microseconds at each attenuation step, with a typical switching time in the order of hundreds of nanoseconds (see individual model datasheets for precise specifications).

## 21. What attenuation level will the RUDAT / RCDAT programmable attenuators power up with?

By default the attenuators will start up with the maximum attenuation loaded but this setting can be changed by the user, either programmatically or using the provided GUI. The start-up options available are:

1. Load the maximum specified attenuation value
2. Load last remembered attenuation value
3. Load a specific attenuation value

In cases 2 and 3, the attenuator will momentarily assume the maximum attenuation state on start-up before reverting to the user specified state.

## 22. Can I set the attenuation of a RUDAT attenuator and then disconnect the computer?

It is possible to set the attenuation and then disconnect the computer control connection while the last set attenuation state is retained, in the following scenarios:

1. **Control via USB, DC power via the RS232 interface.** Recent RUDAT attenuator models can draw DC power through pin 1 of the RS232 connector (see table 3 for applicable models). As long as this remains connected then the USB cable can be disconnected from the controlling PC and the attenuator will retain its last state.
2. **Control via RS232, DC power via USB.** The attenuator can be powered through the USB interface from a powered USB hub or from the AC mains using the USB-AC\DC-5 adapter. The RS232 connection can then be used to control the device from a PC and can be safely disconnected without affecting the attenuation.
3. **Pre-defined "power-up state".** The attenuator can be configured to assume a pre-defined attenuation level when the DC power is applied, either the last remembered attenuation or a user-specified value. When this is set, the DC power supply can be applied through either the USB interface or pin 1 of the RS232 connector (see table 3 for applicable models) and a known attenuation will be loaded, without the need for a controlling PC.

Model Name	Serial Number Range
RUDAT-6000-30	11405010010 or later
RUDAT-6000-60	11407150001 or later
RUDAT-6000-90	11403230000 or later
All other RUDAT models	All serial numbers

Table 3 - Summary of RUDAT models supporting DC power over RS232 connector pin 1

## 23. Can I set the attenuation of a RCDAT attenuator and then disconnect the computer?

It is possible to set the attenuation and then disconnect the computer control connection while the last set attenuation state is retained, in the following scenarios:

1. **Control via Ethernet, DC power via USB.** The attenuator can be powered through the USB interface from a powered USB hub or from the AC mains using the USB-AC\DC-5 adapter. The RJ45 connection can then be used to control the device over the network and can be safely disconnected without affecting the attenuation.

**Pre-defined "power-up state".** The attenuator can be configured to assume a pre-defined attenuation level when the DC power is applied, either the last remembered attenuation or a user-specified value. When this is set, the DC power supply can be applied through the USB interface and a known attenuation will be loaded, without the need for a controlling PC.

## 24. Can the RCDAT attenuator series be controlled without using a DLL?

The simplest method for control in a Windows environment when connected over USB is to use the DLL file. However, if you are able to use the attenuator's RJ45 connection, rather than USB, then no external libraries (including DLL files) are needed. The attenuator has a simple ASCII string command set that can be sent using HTTP or Telnet. In the simplest case you can use any Internet browser (eg: Internet Explorer or Google Chrome) to communicate with the attenuator by typing the commands in the address bar, for example (where 192.168.9.73 is the IP address of an attenuator on the same network as the controlling computer):

<code>http://192.168.9.73/:SN?</code>	Get serial number
<code>http://192.168.9.73/:MN?</code>	Get model name
<code>http://192.168.9.73/:ATT?</code>	Read attenuation
<code>http://192.168.9.73/:SETATT=20</code>	Set attenuation to 20 dB
<code>http://192.168.9.73/:SETATT=0.25</code>	Set attenuation to 0.25 dB



## 25. How are power sensor offset values interpolated?

When specifying an offset table for the power sensor using the DLL, what offset value is used when measuring at a frequency that isn't specified in the table? For example, if the offset table is as follows:

- Frequency = {500, 1000, 2000}
- Offset = {0.5, 2.5, 4.5}

What offset value will be used when measuring at the below frequencies (ie: when the power sensor compensation frequency is set to the below values)?

1. 1 MHz?
2. 501 MHz?
3. 750 MHz?
4. 6000 MHz?

Linear interpolation is used for frequencies between the offset table entries and the minimum / maximum values in the table are used for frequencies outside the range of the table. So, in the example above, the offsets used are:

1. 1 MHz => 0.5 dB
2. 501 MHz? => ~0.5 dB
3. 750 MHz? => 1.5 dB
4. 6000 MHz => 4.5 dB

## 26. How can I control my PWR-SEN-6G(+) power sensor in Windows 7?

PWR-SEN-6G+ units with serial numbers before 11010000000 have an older hardware revision that requires installation of an additional USB driver. This model and the drivers (obtainable from the Mini-Circuits website) are only supported up to Windows XP however it may still be possible to install the driver successfully on 32-bit versions of Windows 7. Please note that due to the age of hardware and driver, they are not compatible with 64-bit operating systems.

All subsequent Mini-Circuits power meter models (for example PWR-6GHS) are USB HID (Human Interface Devices) which are automatically recognized across a range of platforms (both 32 and 64 bit) without the need for additional driver installation.

## 27. Can the PWR series power sensors interface with other test equipment by USB?

Yes, using the USB interrupt codes provided in the programming manual for Linux support, a user can prepare their own interface for any test equipment which supports USB host and USB HID. See the programming manual for full details.

## 28. Which SPI modes are supported in the RS232/USB-SPI converter GUI?

Mini-Circuits' RS232/USB-SPI converters include chip-select (CS), data-in (DI), data-out (DO), clock (CLK) and load enable (LE) connections. The GUI allows the below options:

- CS = ticked: CS set to logic 0, then SPI data is sent, then CS is set to logic 1
- CS = unticked: CS not used
- LE = ticked: SPI data is sent, then LE set to logic 1, then LE set to logic 0
- LE = unticked: LE not used

## 29. What is the purpose of the LE connection on the RS232/USB-SPI converter?

The basic SPI interface has four signals: Data In Data Out, Clock and Chip select. What is the function of the additional Latch / Load Enable (LE) connection and is it needed for SPI communication?

The fifth signal (Load Enable) isn't really a part of the SPI interface at all but these converters have it to support some additional applications. If you're just interested in SPI communication then you can ignore it. The main application using the LE connection is with our ZX76 digital step attenuators which accept serial control data into an internal shift register and then LE is toggled high to load the attenuation. You could also use the LE connection as an additional signaling path if there is any need in your application, in parallel to the SPI communication. The DLL contains the functions Set\_LE to freely set the pin to logic low/high as required and Send\_Receive\_SPI (to send and receive SPI data according to the usual standards) also includes an option to set the LE pin after transmission is complete.

## 30. How do I send more than 16 data bits using the RS232/USB-SPI converter?

My SPI device requires an 8 bit instruction code followed by a 16 bit address, for a total of 24 bits. The programming manual for your USB/RS232 to SPI converter notes that the maximum number of data bits is 16. How do I send/read 24-bits of data?

The simplest method is just to send the SPI commands twice, sending/requesting 16 bits, then the final 8 bits, ie:

```
Send_SPI(16, first_data_part)
Send_SPI(8, second_data_part)
Receive_SPI(16)
Receive_SPI(8)
```

### 31. When should I manually set the input range parameter of the frequency counter?

In general the Mini-Circuits' frequency counters can be left in "auto" measurement range mode where they will automatically choose the most efficient settings for measuring the frequency of the input signal. The process followed by the frequency counter is:

1. Adjust input attenuator to provide optimal signal level
2. Identify correct frequency range
3. Sample signal (this is the sample time set by user)
4. Display measured frequency

Manually setting the frequency range eliminates step 2 which can offer a very slight advantage in measurement speed.