WXScheduler

Category: Automation,Projects,WiRES-X
September 22, 2025
Update: WXScheduler 2.9.9 is available. The 2.9.9 installer is here.

Here’s what has changed with 2.9.9:

WXscheduler v2.9.9 updates: Several fixes related to missing automation delays that are required for properly updating Wires-X Settings.

WXscheduler v2.9.8 updates: – Now restoring Wires-X app to the Desktop only when Wires-X is currently minimized and WXscheduler is about to perform automated Wires-X Settings updates. – If the Wires-X app was minimized at the start of automated Wires-X updates, then the Wires-X app will be minimized again after the automation actions have been completed. – Validation of 5 digit “Return to RoomID” having an even first digit, has been removed.

WXscheduler v2.9.7 provides many improvements. The key ones are: – Compatible with Wires-X versions 2.010 and v1.5xx – Faster automation of the scheduled Wires-X Settings updates that are independent of CPU speed and background load. – Replaced main window’s “Scheduler list” with “View/Modify Schedule” button. – “Wait for Wires-X.exe to Restart” functionality that maintains a valid automation connection to Wires-X. Additional information about WXscheduler can be found in the installation folder: WXsched-info.txt WXsched-license.txt WXsched-Python.txt WXsched-updates.txt WXscheduler.pyw (source code) for those who still want to run it as Python program. All I ask is that you pass along any improvements to w9lbr@arrl.net for the benefit of the amateur radio community.

2.9.1 fixes problems with some Windows installations not working because there no longer is a “Programs (x86)” directory.

WXScheduler is a program that allows WiRES-X nodes to automatically switch to a Room or Node at a predetermined date and time. It’s written in Python, but we now have a Windows installer which does not require the installation of Python.

Description

WXScheduler is a program that allows your WiRES-X node to be connected to another Room or Node at a predetermined day of week, hour and minute in the specified timezone. If your browser balks downloading WXscheduler-v290-Install.exe and you cannot override the warning, then rename the xxxxx.crdownload file using File Explorer or CMD. NOTE: Recent Windows Security updates may rightly so quarantine WXscheduler-v290-Install.exe. Detecting it as a Trojan. Well that is correct only on the basis that it will install another EXE file You can override this by going into Windows Security –> Protection History and find the most recent “threat” and click the Restore link. Finally this especially occurs if you are using Windows’ OneDrive. Which I highly recommend that you do not use! WXscheduler is written in Python, converted into a Windows EXE, and delivered as an installable Windows application for Windows-7 or higher. Which means the User no longer has to install Python and library packages to run WXscheduler.pyw. Additional information about WXscheduler can be found in the installation folder: WXsched-info.txt WXsched-license.txt WXsched-Python.txt WXsched-updates.txt WXscheduler.pyw (source code) for those who still want to run it as Python program. All I ask is that you pass along any improvements to w9lbr@arrl.net for the benefit of the amateur radio community.

The 2.9.0 update changes:

-Address varying CPU speeds and their impact on the automation

-Fixes the problem when AccHistory.log contains garbage in a location string, causing a non recoverable exception. Now the corrupt string is displayed in place of the 6 character GridSquare string.

The old 2.6.1 installer is available here. The Github repository is here.

If you experience problems, please use the Fusion Technical Net to report them (preferred), via Github issues, or email.

Revision History:

WXScheduler was written by Bill, W9LBR. It has been packaged as an .exe with a Windows installer by Chris K9EQ. In addition, the following changes have been made.

-2025-09-15 Address varying CPU speeds and their impact on the automation

Fixes the problem when AccHistory.log contains garbage in a location string, causing a non recoverable exception. Now the corrupt string is displayed in place of the 6 character GridSquare string.

– 2023-02-12 2.6.1a Clicking the ‘x’ button on the scheduler menu no longer causes the program to crash

– 2023-01-10 2.5.1 Removed radioID error messages from the log (there were too many of them and they didn’t add value)
Added an icon for the program

The installer creates a WXScheduler directory under the WIRESXA directory where the Yaesu software lives. It installs the following files:

– Readme.txt, this text
– WXScheduler.pyw, the Python source file
– WXScheduler.ico, the icon file
– WXScheduler.exe, the executable
– Uninstal.exe, the program to uninstall WXScheduler

The uninstaller removes these files and the directory. It does not remove the .json
configuration file one directory above.

The packaged version displays the GUI without a shell console. If you want the console as well,
rename the file to *.py instead of *.pyw. You'll then need to open up a shell (command prompt)
and launch with a 32-bit Python interpreter as in: "C:\python32 WXSheduler.py".
The source code is available at https://github.com/K9EQ/WXScheduler.
Also see HamOperator.com for more information about WXScheduler.

Note that the WXScheduler is either stored in the \OneDrive\Documents\WIRESXA folder, if it exists, or in the user's Documents\WIRESXA folder. So if you can't find it in your WIRESXA folder, you may have OneDrive so look there.

WXscheduler v2.5 updates:
– Schedule settings now require a Timezone parameter.
– Compensation for Daylight Savings Time is now automatic on a per timezone basis
– Load Python Timezone package using Windows Command Prompt:
pip install pytz
– WXscheduler.cfg is now more portable and user editable
– User's HOME path is now resolved at run time and no longer saved in WXscheduler.cfg

– JSON data fields are now saved in pretty mode
– When WXscheduler-v2.5 detects an existing WXscheduler.cfg without timezone data fields,
the user will be prompted to set their local timezone value and all existing events will
be updated with the local timezone value.
– WXscheduler's main window:
– Scheduled events are displayed in chronological order
– When the main window is moved, its new position is remembered. And all sub-windows
will be opened in the same position. This replaces PySimpleGUI's default of centering
window within the display.
– [Add Event] and [Delete Event] buttons have been replaced with the [Scheduler] button that
when clicked provides a new window with buttons: [New] [Delete] [Update] [Cancel]

WXscheduler v2.4 updates:
– Additional exception debug information
– Changed main window title from "Wires-X Scheduler" to "WXscheduler (v…)"

WXscheduler v2.3 update:
At startup determines whether Microsoft OneDrive is active or not and locates
the Documents/WIRESXA and user/Desktop folders accordingly.

WXscheduler v2.2 updates:
At startup verifies:
– Wires-X.exe is accessable at its standard location
– /Users/????/Documents/WIRESXA folder is accessable (where WXscheduler.cfg is stored)

WXscheduler v2.1 updates:

Better displays exception information
Shows expected WIRESXA pathname to last heard data file

WXscheduler v2.0 update adds:
Display Last Heard information that the Wires-X application updates once a minute.
If available, Lat/Lon displayed as 6-character Grid Square.

Wxscheduler v1.3 updates:
Last Heard information now correctly displays the newer Yaesu models
(i.e. FT5-D, FTM-200, and FTM-500)
Desktop\Wires-X_Last_Heard.html contains hypertext encoded callsigns
that perform a QRZ.com callsign lookup.

Prerequisites (versions below are what was tested on):

Windows PC (win7 and higher)
Wires-x App (Ver-1.550)
Python 3 (32-bit) because Wires-X App uses the win32 UI
site-packages: PySimpleGUI, pywinauto, pytz

Installing Python:

Recommend going to https://www.python.org/downloads/windows/
to find a Python release that matches Windows on your PC.

After installing Python, use a CMD window to load site-packages:
– pip install pywinauto PySimpleGui pytz

Installing WXscheduler.pyw:

Copy WXscheduler.pyw to your Desktop

Double click on the WXscheduler.pyw icon to launch the program.

Hint: To see any startup or run time issues:

– Open a CMD window
– Copy WXscheduler.pyw WXscheduler.py
– WXscheduler.py

WiRES-X Automation

Category: Automation,Projects,WiRES-X
September 22, 2025
Yaesu does not provide a mechanism that allows the WiRES-X software to be controlled by another program, i.e., having another program switch to a certain Room when a net starts.

Windows does, however, permit another program to send events to a program. Each window, menu item, and dialog in a Windows program has a unique identifier. It is possible to use these identifiers to send “message” to the WiRES-X software.

The WiRES-X Automation Project’s purpose is to bring together people who are interested in developing this technology and sharing their results.

To get things going, here are two mechanisms for automating WiRES-X:

1. AutoIT: http://www.autoitscript.com
2. Python – an excellent programming language found at python.org

Dave, N9TOW, provided the following information:

Packages I have installed on my WiresX system.

```
C:\Users\WiresX>pip list
comtypes (1.1.3)
pip (9.0.1)
pypiwin32 (220)
pywinauto (0.6.3)
setuptools (28.8.0)
six (1.10.0)
https://github.com/pywinauto/pywinauto
```

To install

pip install -U pywinauto

Script that executes changing channels on WiresX app

```
import time
from pywinauto import Application
  app  =  Application().connect(path="C:\Program  Files (x86)\YAESUMUSEN\WIRES-X\wires-X.exe")
app.WiresX.menu_select("Connect(C)->Connect To(T)")
time.sleep(1.5)
app.InputID.Edit.set_edit_text("21493")
time.sleep(.5)
app.InputID.OK.click()
time.sleep(4)
dialogs = app.windows()
##app.Dialog.CloseButton.click()
```

Python Program

Bill, W9LBR, developed a Python program that runs on the WiRES-X computer. The software has been updated as of 12-Feb-2024. See more here.

This is from his description:

WXscheduler.pyw is a Python 3 program that automates certain aspects of
Wires-X Room/Node connections according to a userrdefined schedule.
To that end WXscheduler.pyw must be run on the same Windows PC that hosts the Wires-X application controlling either an HRI-200 or
a USB connected radio that supports PDN operation.

Schedulable operations:
– Connect
– Disconnect
– SetUnlimitedTOT
– SetTimeoutTOT
– Restart Wires-X App

Manual operation: Button that causes a Force Disconnect

Bonus operation: Displays Last Heard information that the Wires-X app
updates once a minute. Converting Lat/Lon into Grid Square if available.

A zip file containing instructions and the Python program is available at WXScheduler.

Update History

2 June 2023 v1.3 updates: – Last Heard now displays FT5-D, FTM-200, and FTM-500 correctly – Desktop\Wires-X_Last_Heard.html now generated to be viewed with PC browser

12 Feb 2022 updates: – Corrected grid square rendering of GPS coordinates as displayed in the Last Heard frame – version (v1.2) now displayed in main window title

5 Feb 2022 minor updates: – Fixed Last Heard display of recently

manufactured FT70-D radios – WXscheduler.txt is now in DOS format, making it easier to read with Notepad 2022 version of WXscheduler.pyw – A scheduled event now performs multiple actions – Added File->Settings->Call settings actions – Fixed issue when Wires-X App is minimized – Fixed issue when Wires-X Settings window left open