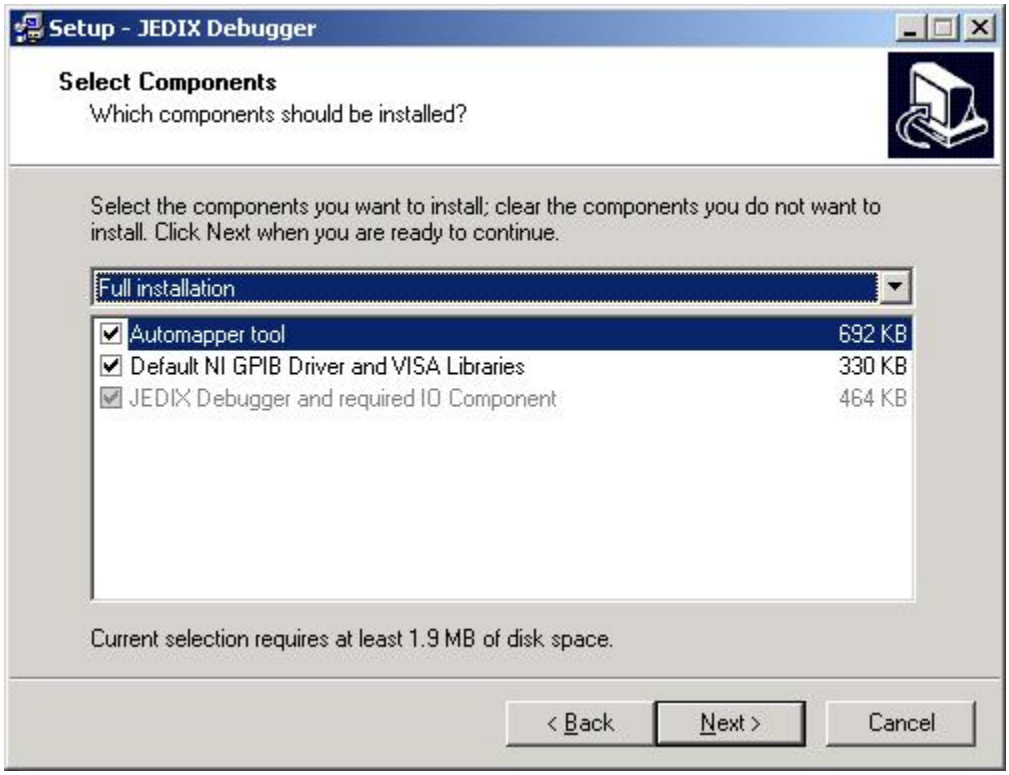
ReadMe
JEDIX Debugger v1.1.0.0
Installation and Usage Notes

Installation

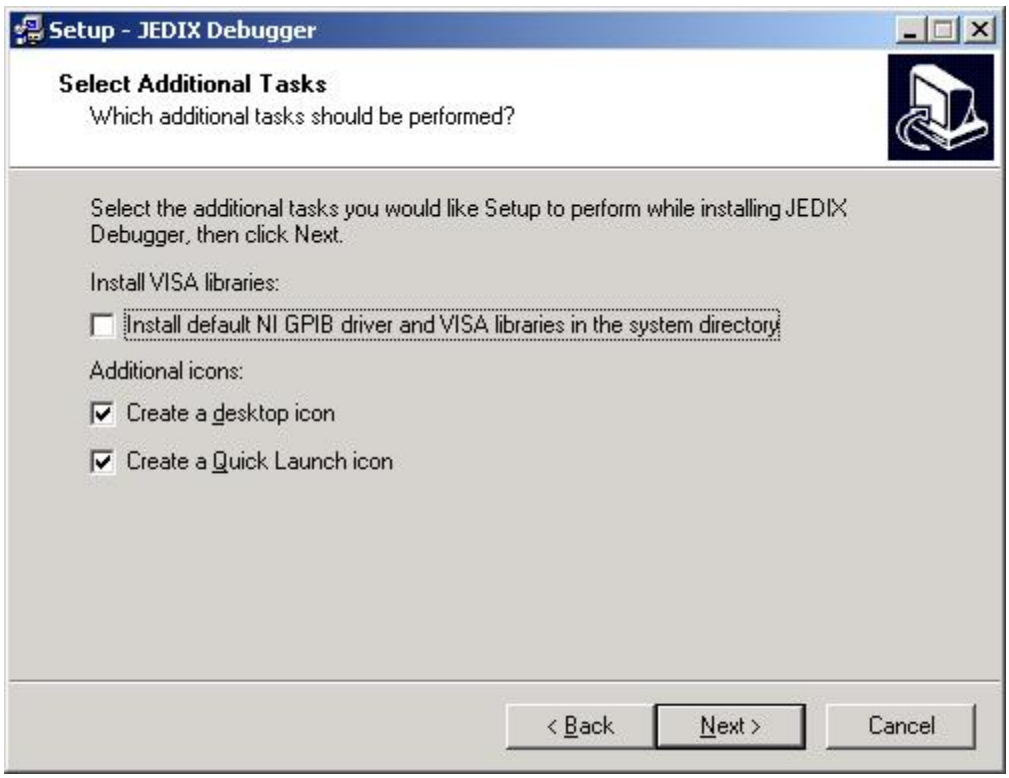
Open the attached file JediX_Setup.abc – You will need to change the file extension form .abc to .exe to run the installation.

Proceed with the installation while following these important instructions:

On the screen titled “Select Components” choose “Full installation” from the combo box:



Press “Next” to continue.

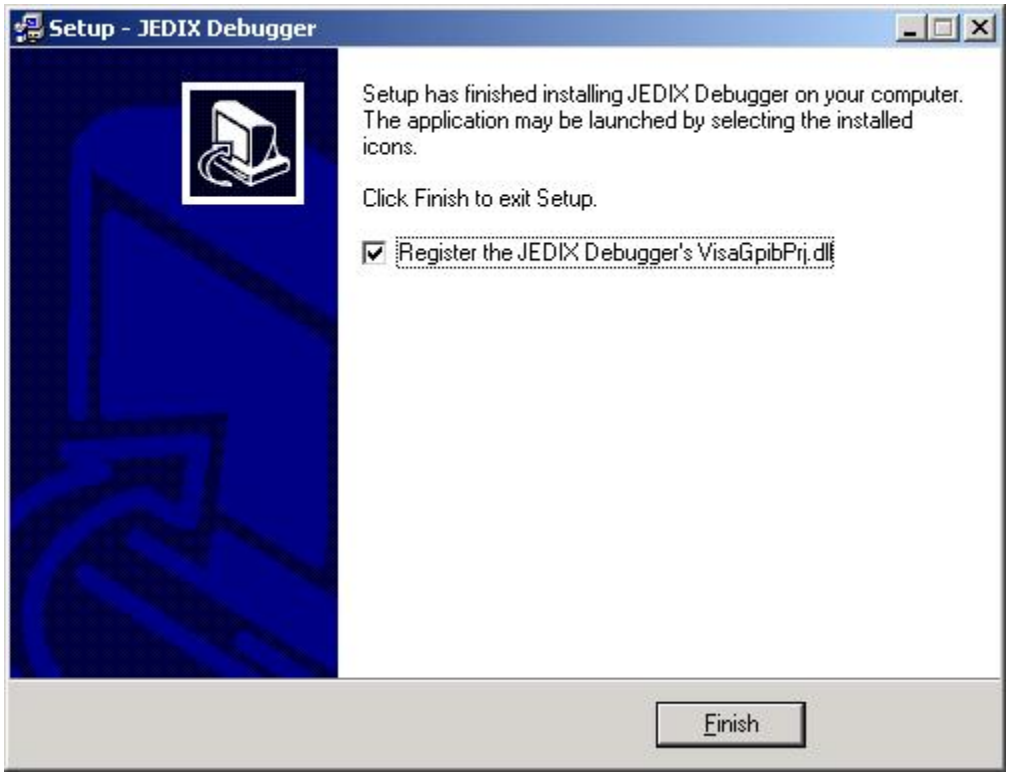


Be sure that you **UNCHECK** the option titled "**Install default NI GPIB Driver and VISA libraries in the system directory**" if you have a GPIB card installed in the computer.

This option is only used for computers that you will NOT be installing a GPIB card in. If you leave this option checked and you have a GPIB card then your card may no longer function correctly.

Press "Next" once you've chosen the correct options.

Proceed with the installation by pressing "Next" and then "Finish" until you are presented with the following window:



For JEDIX to function correctly, this driver must be registered. You can let this installation do it for you now by leaving the box checked, or you can manually register it later.

If you need to manually register the file, you can find it under the default installation directory *C:\Program Files\JEDIX Debugger\Driver\RemoveVisaGpibPrj.dll*

To register, use regsvr32.exe and give it the path to the file on your system.

Press “Finish” to complete the installation.

JEDIX in a Nutshell (a short description)

JEDIX allows you to:

- Show all communication on GPIB bus in real-time for TME.
- Have real-time control of any test
- Persist GPIB communication to a file.

These simple abilities allow us to do a lot more than we've been able to do in the past.

How does JEDIX work?

- Uses "drop in" replacement for Visa IO Component (A modified version of VisaGpibPrj.dll).
- JEDIX interface and Modified IO Component communicate together using inter-process communication and shared memory.
- Modified IO component tells JEDIX asynchronously when an "event" is occurring and provides the data associated with the event.
- This asynchronous communication allows the JEDIX to capture and process all GPIB data sent from the Modified IO Component in the correct order.

Getting Started

Under your start menu is a section called Programs\JEDIX Debugger. Choose JEDIX Debugger to launch the program. Please note that JediX will inform you if the necessary IO component is not registered. Refer to the installation instructions for notes on how to register this dll.

Taking Control

You can test that JEDIX is working correctly by trying to detect a DUT with TME while JEDIX is running.

Here's how to do this:
If not already started, load JediX from the start menu or shortcuts created after install. Under the "Window" menu, choose "Show/Hide trace Window" so that a new window appears at the bottom of the JediX window. (This window just allows you to see the bus commands without having to actually record them to a file). Go back to TME and configure your DUT. If things are working correctly, you will see some GPIB commands in the bottom "Trace Window" of JEDIX after you detected the DUT. If preferred, you can choose the same option under the "Window" menu to hide the Trace Window.

Recording a test

Recording a test allows you to persist a test to file for later use. Some uses include being able to view what happened in the test, attaching the test to an ATM, searching the test for specific commands or values, and using the test for playback. Recorded tests can also be modified to playback different values to see how the software responds.

Open up JEDIX (if not already) and create a new file by clicking the new file icon or choosing "New" from the file menu. A new document will be created for you.

With your document selected, press the red circle icon or choose "Start Recording" from the Recording menu. While recording, anything that occurs on the GPIB bus will automatically show up in the document that you're recording to (as well as the trace window).

To stop recording, press the red circle icon, black square icon, or choose "Stop Recording" from the Recording menu.

Note: You cannot save a file while you are recording.

Once you've finished recording, press the save icon or choose "Save" from the file menu to save the recording to your computer.

Note: You cannot playback and record from the same file.

Known Issues/Concerns

The "Time" column does not always display correct or expected values. This is a known issue and portions of it cannot be resolved due to design. (It makes sense when you think about playback vs. recording for a bit). However, in some circumstances the information is valuable enough to justify the times where it's not. Specifically, during playback of a file some commands in the trace window will show the time of recording and some will show the time of playback.

Some tests don't playback as expected, or the data changed unexpectedly. One example of this are the timeout values that given to the power meters. For some reason the timeout values are different during playback of a test than when they were recorded.

Another way that tests playback differently than when recorded occurs when the driver throws an exception but the test is allowed to continue. During initial design of JediX, I incorrectly assumed that any exception thrown in the driver would result in an aborted test. Because of this, I did not design JediX to correctly deal with exception conditions. What happens is that during recording an exception will get thrown from the driver and JediX will still capture the command correctly. But during playback, no exception will ever get thrown. Since the driver no longer throws an exception, the test will now start playing back differently. This situation was found in PSG for the DoPUP() method of the PSG driver. The workaround was to modify the recorded file by deleting the extra

commands where exceptions were thrown (multiple “*OPC?” commands in a row) so that there was only one command.

I am still researching everything I can to determine what impacts a test in regards to recording and playback differences. I’m sure there are some I’ve missed and a process will have to be created/modified to deal with these.

Additionally, not all functionality has been implemented. You may see a few messages here or there that talk about missing functionality.