

## INTRODUCTION

This document discusses the installation process of the 1-Wire<sup>®</sup> drivers for Microsoft Windows<sup>®</sup> 32-bit operating systems. The software application developer has two options to accomplish a 1-Wire drivers installation. The first is using Dallas Semiconductor's "1-Wire Drivers" installation program, and the second is developing a custom installation program.

## DALLAS SEMICONDUCTOR'S 1-WIRE DRIVERS INSTALLATION PROGRAM

The easiest option for a 1-Wire software developer is to use the installation program created by Dallas Semiconductor. Also known as the TMEX RTE (runtime environment), the "1-Wire Drivers" installation program performs a comprehensive install of all 1-Wire adapter drivers, including the associated application programming interface (API) libraries, a COM object, its respective classes, and optional .NET support. It also writes the necessary registry keys for the software's proper functioning. The "1-Wire Drivers" installation program is available for download from the iButton<sup>®</sup> website: <http://www.ibutton.com/software/tmex>.

The developer that wants to distribute a 1-Wire application has a couple of choices in using the "1-Wire Drivers" installation program. One choice is to notify the end customer to download and install the "1-Wire Drivers" installation package. This is the easiest method, but it involves the end customer installing two separate software pieces. The first software piece is the "1-Wire Drivers" installation program and the second is the software developer's 1-Wire application. Another choice for the software developer is to make a custom installation program that not only installs the developer's own program but spawns the "1-Wire Drivers" installation program as well.

## CREATING CUSTOM INSTALLATION PROGRAMS

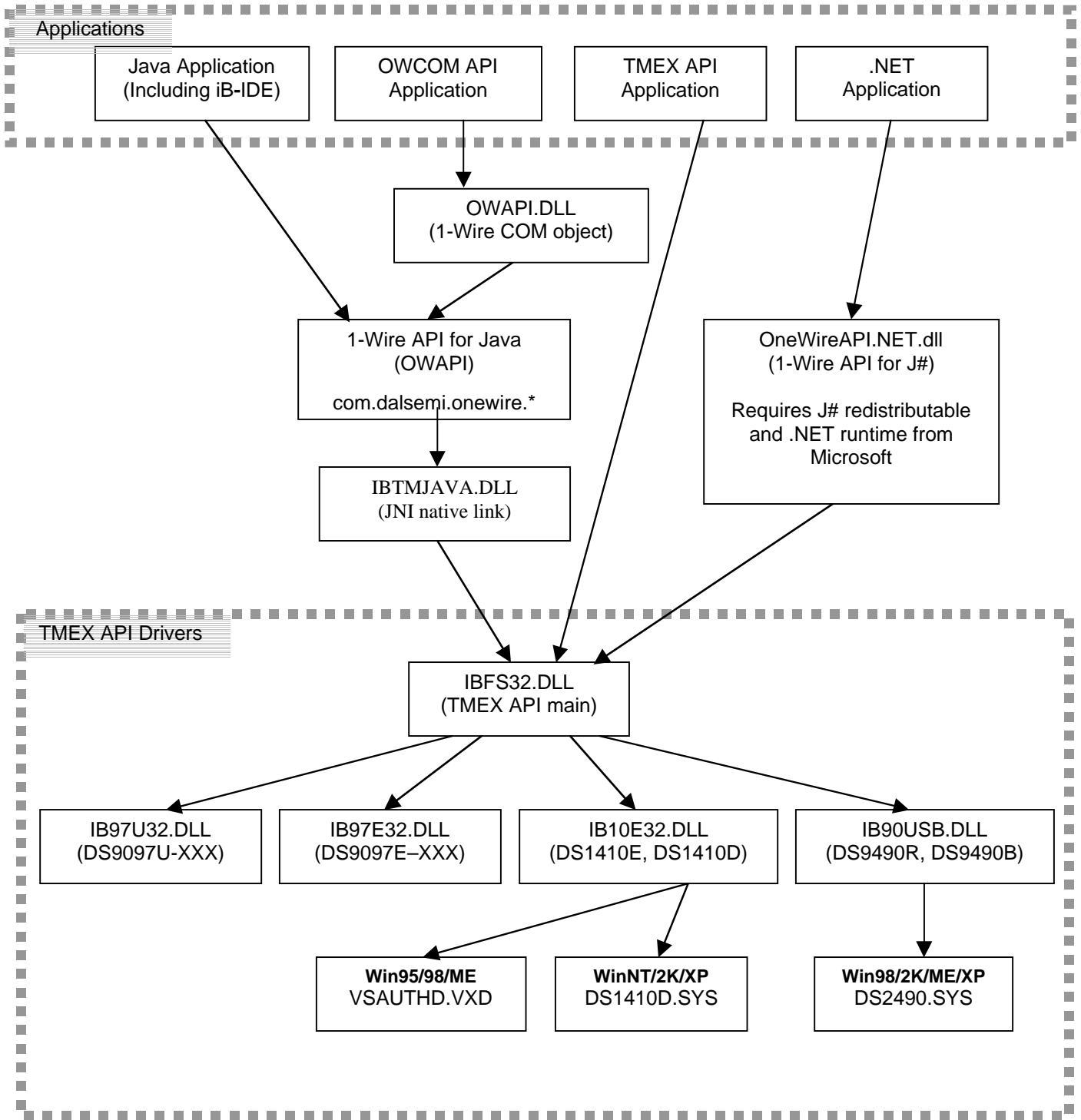
Some 1-Wire software application developers either desire to or are required to make a single installation program that includes their custom 1-Wire application and the necessary 1-Wire drivers and other API libraries. Of these developers, many do not need the entire "1-Wire Drivers" installation, but only the drivers necessary for their software application. To make a custom installation of drivers, the software developer needs to understand the inter-dependencies of the different 1-Wire drivers and API libraries. Figure 1 below illustrates the API and driver software dependencies.

Determining which files are needed for a custom installation requires the developer to know which API is being used and which hardware adapter(s) will be supported by the application. The API used could be either a derivative of the 1-Wire API for Java<sup>™</sup> (such as the OWCOM API or our .NET support) or the TMEX API. For a more complete discussion of both APIs and a list of hardware adapters supported by the 1-Wire Drivers, please see the documentation included in the 1-Wire SDK for Windows available for download here: <http://www.ibutton.com/software/tmex>.

Table 1 lists the files potentially involved in a 1-Wire Drivers custom installation, along with API and adapter dependencies. A description of each file is listed, along with the destination directory the file should be copied to on the end customer's computer. Please note that variables are used to represent the destination directories, as they can differ for each Microsoft installation. For example, the default windows\system directory depends upon the operating system being used (i.e., C:\windows\system or C:\winnt\system32). These are represented in Table 1 as %WINDSYS%. The windows directory is also represented as a variable, %WIN%. Table 2 lists the appropriate registry entries to complete a custom installation.

*1-Wire and iButton are registered trademarks of Dallas Semiconductor.  
Windows is a registered trademark of Microsoft Corp.  
Java is a trademark of Sun Microsystems.*

Figure 1. 1-Wire Drivers and API Connectivity



**Table 1. 1-Wire Drivers File List With Installation Instructions**

| <b>Files</b>   | <b>Description</b>                              | <b>Copy To Destination Subdirectory</b>   | <b>API</b>                  | <b>Microsoft OS Support</b> | <b>1-Wire Adapter Type</b> |
|--|---|---|-----------------------------|-----------------------------|----------------------------|
| <b>OneWireAPI.NET.dll</b>  | .NET interface to the 1-Wire API for J#         | Preferably, copied to end-user application's directory. Requires Microsoft's .NET redistributable and Visual J# redistributable installed on end-user's computer. | .NET (1-Wire API for Java)  | 98, ME, NT, 2000, XP        |                            |
| <b>OWAPI.DLL</b><br><b>(See below for required COM registration process)</b>   | COM interface to the 1-Wire API for Java        | %WIN\SYS%<br>(Note: %win% stands for the system directory, usually C:\Windows\System or C:\WinNT\System32)  | OWCOM (1-Wire API for Java) | 95, 98, ME, NT, 2000, XP    |                            |
| <b>1-Wire API for Java classes (copy entire directory tree to destination)</b> | 1-Wire API for Java                             | %WIN%\JAVA\TRUSTLIB<br>(Note: %win% stands for the windows directory, usually C:\Windows or C:\WinNT)   | OWCOM                       | 95, 98, ME, NT, 2000, XP    |                            |
| <b>IBTMJAVA.DLL</b>  | Java JNI native interface to TMEX API           | %WIN\SYS%   | OWCOM                       | 95, 98, ME, NT, 2000, XP    |                            |
| <b>IBFS32.DLL</b>  | Main TMEX API library                           | %WIN\SYS%   | .NET<br>OWCOM<br>TMEX       | 95, 98, ME, NT, 2000, XP    |                            |
| <b>IB97U32.DLL</b>   | Serial port (DS2480B-based adapter) sub-driver  | %WIN\SYS%   | .NET<br>OWCOM<br>TMEX       | 95, 98, ME, NT, 2000, XP    | <b>Serial</b>              |
| <b>IB97E32.DLL</b>   | Serial port DS9097E (legacy adapter) sub-driver | %WIN\SYS%   | .NET<br>OWCOM<br>TMEX       | 95, 98, ME, NT, 2000, XP    | <b>Serial (legacy)</b>     |
| <b>IB10E32.DLL</b>   | Parallel port DS1410D/E sub-driver              | %WIN\SYS%   | .NET<br>OWCOM<br>TMEX       | 95, 98, ME, NT, 2000, XP    | <b>Parallel</b>            |
| <b>VSAUTHD.VXD</b>   | Parallel port DS1410D/E device driver           | %WIN\SYS%   | .NET<br>OWCOM<br>TMEX       | 95, 98, ME                  | <b>Parallel</b>            |
| <b>DS1410D.SYS</b>   | Parallel port DS1410D/E device driver           | %WIN\SYS%\DRIVERS   | .NET<br>OWCOM<br>TMEX       | NT, 2000, XP                | <b>Parallel</b>            |
| <b>IB90USB.DLL</b>   | USB (DS2490-based adapter) sub-driver           | %WIN\SYS%   | .NET<br>OWCOM<br>TMEX       | 98, ME, 2000, XP            | <b>USB</b>                 |
| <b>DS2490.SYS</b>  | USB (DS2490-based adapter) device driver        | Default installation directory. See "Special Considerations for USB" below.   | .NET<br>OWCOM<br>TMEX       | 98, ME, 2000, XP            | <b>USB</b>                 |
| <b>DS2490.INF</b>  | USB (DS2490-based adapter) inf file             | Default installation directory. See "Special Considerations for USB" below.   | .NET<br>OWCOM<br>TMEX       | 98, ME, 2000, XP            | <b>USB</b>                 |

**Table 2. Registry Entries**

| Microsoft OS     | Files Affected   | Registry Keys   |
|------------------|--|---|
| All Platforms    | IBFS32.DLL<br>IB97E32.DLL<br>IB10E32.DLL<br>IB97U32.DLL<br>IB90USB.DLL | [HKEY_LOCAL_MACHINE\SOFTWARE\Dallas Semiconductor]<br>[HKEY_LOCAL_MACHINE\SOFTWARE\Dallas Semiconductor\iButton TMEX]<br>[HKEY_LOCAL_MACHINE\SOFTWARE\Dallas Semiconductor\iButton TMEX\3.00]<br>"MainDriver"="IBFS32.DLL"<br>"TYPE1"="IB97E32.DLL"<br>"TYPE2"="IB10E32.DLL"<br>"TYPE5"="IB97U32.DLL"<br>"TYPE6"="IB90USB.DLL"<br>"DefaultPortNum"="1" *set by "Default 1-Wire Net" program<br>"DefaultPortType"="6" *set by "Default 1-Wire Net" program |
| NT, 2000, XP     | DS1410D.SYS  | [HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DS1410D]<br>"ErrorControl"=dword:00000001<br>"Group"="Extended Base"<br>"Start"=dword:00000002<br>"Tag"=dword:00000002<br>"Type"=dword:00000001   |
| 98, ME, 2000, XP | DS2490.SYS<br>DS2490.INF   | Appropriate registry keys set by the Windows "Add Hardware" Installer.  |

## INSTALLATION OF .NET FILES

.NET 1-Wire software requires that the 32-bit Windows operating system on which it runs has the .NET runtime installed, along with the Visual J# redistributable. See the table below on where to get these packages.

**Table 3. Microsoft .NET Package Dependencies**

| Package Needed                 | Link  |
|--------------------------------|---|
| .NET Framework                 | <a href="http://msdn.microsoft.com/netframework/downloads/howtoget.aspx">http://msdn.microsoft.com/netframework/downloads/howtoget.aspx</a> |
| Visual J# .NET Redistributable | <a href="http://msdn.microsoft.com/vjsharp/downloads/howtoget.asp">http://msdn.microsoft.com/vjsharp/downloads/howtoget.asp</a>             |

The .NET 1-Wire support is a Microsoft Visual J# compiled version of the 1-Wire API for Java. The result of which is a single DLL file, OneWireAPI.NET.dll. Installation of this file consists simply of copying it to the folder that the parent application (that depends upon its presence) resides. Thus, a developer only needs to copy it to the folder to which his 1-Wire .NET program gets copied.

The 1-Wire Drivers installation provides .NET support through an optional install program (OneWireAPI.NET\_Setup.msi) that gets copied to the application directory. When double-clicked, the install provides the above-mentioned OneWireAPI.NET.dll, along with example programs written in C#.

Besides the OneWireAPI.NET.dll, the developer must also install the TMEX API files and drivers. For complete instructions, see section below entitled "INSTALLATION OF TMEX API FILES AND DRIVERS".

### Optional .NET MSI Install

The .NET Microsoft Installer (MSI) file that gets installed with the Version 4.00 beta 4 (and up) version of 1-Wire Drivers is actually a small Software Development Kit (SDK). When the file is run (double-clicked), the user chooses which directory the contents of the SDK get placed (otherwise known as the "application directory"). The main contents of the SDK are two C# programs, along with the OneWireAPI.NET.dll. See the table below for a list of files that get installed. Note that the %APP% directory refers to the "application directory" and defaults to the following directory: C:\Program Files\Dallas Semiconductor\OneWireAPI.NET.

**Table 4 File List for Optional .NET MSI Install**

| Destination Subdirectory      | Files  | Description   |
|-------------------------------|--|---|
| %APP%                         | HygrochronViewer.exe   | An application for the DS1923 Hygrochron iButton  |
|                               | HygrochronViewer.exe.config  | Config file for the hygrochronviewer.exe program.   |
|                               | OneWireAPI.NET.dll   | The 1-Wire API for Java, compiled using Visual J#, in a .NET assembly DLL. Add a reference to this assembly to a .NET project to use classes from the 1-Wire API. |
|                               | owio.exe   | General Purpose utility for arbitrary 1-Wire communication.   |
|                               | readme.txt   | Readme file containing basic info on installation.  |
|                               | scpl.dll   | Charting object used by HygrochronViewer to produce graphs.   |
| %APP%\HygrochronViewer Source | app.config<br>AssemblyInfo.cs<br>DevicePasswordForm.cs<br>DevicePasswordForm.resx<br>HygroChronViewer.cs<br>HygrochronViewer.csproj<br>HygroChronViewer.resx<br>StartMissionForm.cs<br>StartMissionForm.resx<br>TemperatureCompensationOptions.cs<br>TemperatureCompensationOptions.resx | C# source code and project files for the HygrochronViewer.exe program.  |
| %APP%\owio Source             | AssemblyInfo.cs<br>owio.cs<br>owio.csproj<br>owio.resx   | C# source code and project files for the owio.exe program.  |

## INSTALLATION OF OWCOM API FILES

OWCOM requires the TMEX API to also be installed. Like the .NET support, the TMEX API is used as a foundation on which to build a higher level API. To successfully install the OWCOM API files, the developer will need to perform the following steps:

1. [Highly recommended] Look for the presence of the Microsoft Java Virtual Machine (JVM) on the end-user's computer and notify the end-user where the JVM can be downloaded if it does not exist. One useful download site of Microsoft's JVM is here:  
<http://www.big-foot.de/dateien/Msjava.exe>
2. The JVM is not installed by default on some versions of Windows 95 and Windows XP. A simple way to detect the presence of the JVM is to look for the existence of the file Jview.exe (Microsoft's Java interpreter). It is normally found in either the Windows directory or the Windows\System directory. Please see Code Example 1 below for example pseudocode on how to find Jview.exe.

**Code Example 1. Finding Microsoft's JVM**

```
// Example pseudocode snippet to find Microsoft's JVM (Jview.exe)
// If Jview.exe not found, then display an appropriate warning message
JVMPresence Boolean; // Variable to determine if JVM is present
FileNameWindows String; // Variable to hold path to the windows directory
FileNameSystem String; // Variable to hold path to the system directory

JVMPresence = False; // Initialize variable to False
FileNameWindows = GetWindowsDir + "\\JView.exe"; // Get "windows" directory path
FileNameSystem = GetSystemDir + "\\JView.exe"; // Get "system" directory path

// check for existence of jview.exe in Windows and/or System directory.
if FileExists(FileNameWindows) then JVMPresence = True;
if FileExists(FileNameSystem) then JVMPresence = True;

// if file not found, show warning message
if (not JVMPresence) then
  Application.MessageBox("WARNING, the Microsoft Java Virtual Machine (JVM) has
  not been detected on this machine. The 1-Wire COM object will be unavailable
  until it is installed. The JVM can be downloaded from:
  http://www.big-foot.de/dateien/msjava.exe");
```

3. Copy the 1-Wire API for Java files to their destination directories. Please copy the entire file tree to the following directory: %win%\JAVA\TrustLib. Note that %win% stands for the windows directory. Here is a list of 1-Wire API subdirectories that will need to be copied to %win%\JAVA\TrustLib:

```
com\dalsemi\onewire\*.class
com\dalsemi\onewire\adapter\*.class
com\dalsemi\onewire\application\file\*.class
com\dalsemi\onewire\application\monitor\*.class
com\dalsemi\onewire\application\sha\*.class
com\dalsemi\onewire\application>tag\*.class
com\dalsemi\onewire\container\*.class
com\dalsemi\onewire\debug\*.class
com\dalsemi\onewire\utils\*.class
nanoxml\*.class
nanoxml\sax\*.class
org\xml\sax\*.class
org\xml\sax\helpers\*.class
```

4. Copy the Java JNI library, ibtmjava.dll, and the COM object, owapi.dll, to the %win%\sys% directory. Since owapi.dll is a COM object, it will need to be registered. Most installation packaging programs such as InstallShield® or the Microsoft Installer will automatically recognize that owapi.dll is a COM object and offer to register it upon installation. A batch file can also be used to install the owapi.dll. See Code Example 2 for an example of a batch file installation. Registering and un-registering the COM object can be done manually through the regsvr32.exe program available on Microsoft 32-bit OS platforms. See instructions below.

To register the owapi.dll manually, please type at a command prompt:

```
regsvr32 /s <path to dll>
```

To un-register the owapi.dll, please type at a command prompt:

```
regsvr32 /s /u <path to dll>
```

\*special note: On Win95/98/ME, regsvr32 may have to be run from the %win%\sys% directory.

**Code Example 2. Installation Batch File For OWAPI.DLL**

```

REM - To use this batch file, put the owapi.dll and this batch file
REM - in the same directory, and then run this batch file.
REM - Start of installation.
echo off
set ostyle=9598ME
if %OS%!=Windows_NT! set ostyle=NT
if %ostype%==NT GOTO WINNT
if %ostype%==9598ME GOTO WIN98
GOTO END
:WINNT
copy owapi.dll %windir%\System32
regsvr32 %windir%\System32\owapi.dll
GOTO END
:WIN98
copy owapi.dll %windir%\system
%windir%\system\regsvr32 %windir%\System\owapi.dll
:END

```

5. Install the TMEX API files and drivers as discussed below.

**INSTALLATION OF TMEX API FILES AND DRIVERS**

For a list of all TMEX API files and drivers, along with a description of where these files need to be copied, please see Table 1. Also, look at the necessary registry entries for the associated files and drivers in Table 2.

The TMEX API files and drivers consist of a main dll library, IBFS32.DLL, along with adapter dependent files. This is where a developer can pick and choose which dll files and drivers need to be installed. If a developer only wants DS2480B-based adapters supported (the DS9097U), the custom installation program only needs to install the main TMEX dll library, IBFS32.DLL, and the adapter dependent library file, IB97U32.DLL. The developer's custom install will also need to ensure the appropriate registry entries get written. In this case, from Table 2, "REGISTRY ENTRIES," the registry entries associated with the "All Platforms" row would need to be written.

**SPECIAL CONSIDERATIONS FOR USB**

There are special considerations for installation of the 1-Wire USB files. The developer is encouraged to write the installation program in such a way as to prompt the user with the following messages.

1. Before copying of files takes place, prompt the message: "Unplug any 1-Wire USB devices to allow the device driver to be updated."
2. It is recommended that the USB files be copied to the default installation directory. After files have been copied and the USB inf file has been setup (see below), prompt the following message: "Please plug in the 1-Wire USB device and wait for the Windows 'Add Hardware Wizard' to complete the installation of the USB driver. If driver not immediately found, please point the wizard to the default installation directory". It is a good idea to include in the prompt the default installation directory where the DS2490.SYS and the DS2490.INF files are located. An example of where the 1-Wire Drivers installation package copies the USB driver files to is: "C:\Program Files\Dallas Semiconductor\1-Wire Drivers Version 4.00"
3. If it is not recognized, then reboot and try again. Please note that the USB device driver can be installed or updated by hand. Please see Appendix A under the section entitled "Installing the 1-Wire USB Adapter's Device Driver by Hand".

It is also recommended that the DS2490.INF file be "pre-installed" prior to the user plugging in the USB adapter for the first time. The DS2490.INF file contains the installation instructions that the "Add Hardware" wizard eventually uses to install the DS2490.SYS device driver file. This makes USB device installation go much smoother on Windows 2000 and Windows XP operating systems. To do this requires a simple call to a Microsoft function detailed in the Windows Platform SDK documentation: SetupCopyOEMInf.

Pre-installing the DS2490.INF file requires the following steps be followed:

1. On the target system, create a directory for the driver files. The 1-Wire Drivers package puts them in the default installation directory.
2. Copy the DS2490.INF and DS2490.SYS to the directory created in Step 1.
3. Call SetupCopyOEMInf (described in the Windows Platform SDK documentation), specifying the INF file in the directory created in Step 1.

There are several parameters to the SetupCopyOEMInf function. Specify the SPOST\_PATH constant for the OEMSourceMediaType parameter and specify NULL for the OEMSourceMediaLocation parameter. SetupCopyOEMInf copies the INF file for the driver into the %windir%\Inf directory on the target system (it may change names to OEMXX.inf where XX represents an incremental number) and directs the SetupAPI to store the source location of the INF file in its list of preprocessed INF files. SetupCopyOEMInf also tells the PnP Manager (Plug and Play Manager) to install the driver the next time it recognizes a device listed in the INF file. When the user plugs in the DS9490R/DS9490B device for the first time, the PnP Manager recognizes the device, finds the INF file copied by SetupCopyOEMInf, and installs the drivers.

See Code Example 3 below for an example program written in C that pre-installs the DS2490.INF file. The 1-Wire Drivers MSI (Microsoft Installer package) spawns an external program, chkusr.exe that similarly calls the SetupCopyOEMInf function. The source to chkusr.exe as well as the MSI install is provided in an alternative 1-Wire Drivers download available from: <http://www.ibutton.com/software/tmex>. For further info on pre-installing inf files, please see Microsoft's website (<http://msdn.microsoft.com/>) and perform a site search on the keyword "SetupCopyOEMInf".

### Code Example 3. Pre-Installing DS2490.INF

```

//-----
//
// SetupUSB.c - main purpose is for pre-installing the DS2490.inf file for
//              the DS9490R/B USB Driver
//
#include <stdio.h>
#include <windows.h>
#include <SetupAPI.h>

int main(int argc, char** argv)
{
    char INFFile[256];

    // the inf file should be specified as a
    // command-line option to this program.
    // (i.e., argv[1] + "DS2490.inf")
    memset(&INFFile[0],0x00,256);
    strcpy(&INFFile[0], argv[1]);
    strcat(&INFFile[0], "DS2490.inf");

    SetupCopyOEMInf(&INFFile[0], NULL, SPOST_PATH, SP_COPY_NEWER_ONLY,
                   "DS2490.inf",256,NULL,NULL);

    return 1;
}

```

## CONCLUSION

This document discusses the installation process of the 1-Wire drivers for Microsoft Windows 32-bit operating systems. The target audience is the software developer wanting or needing to perform a custom installation of all or a subset of 1-Wire drivers and API files. If after reading this document, more clarification is needed on the installation of a particular file, or any other software technical support questions arise, the developer is encouraged to post questions to Dallas Semiconductor MAXIM's web-based discussion forum. To subscribe, the developer can go to <http://discuss.dalsemi.com/> and sign up. Engineers at Dallas Semiconductor and other 1-Wire software/hardware developers constantly monitor the web forum.

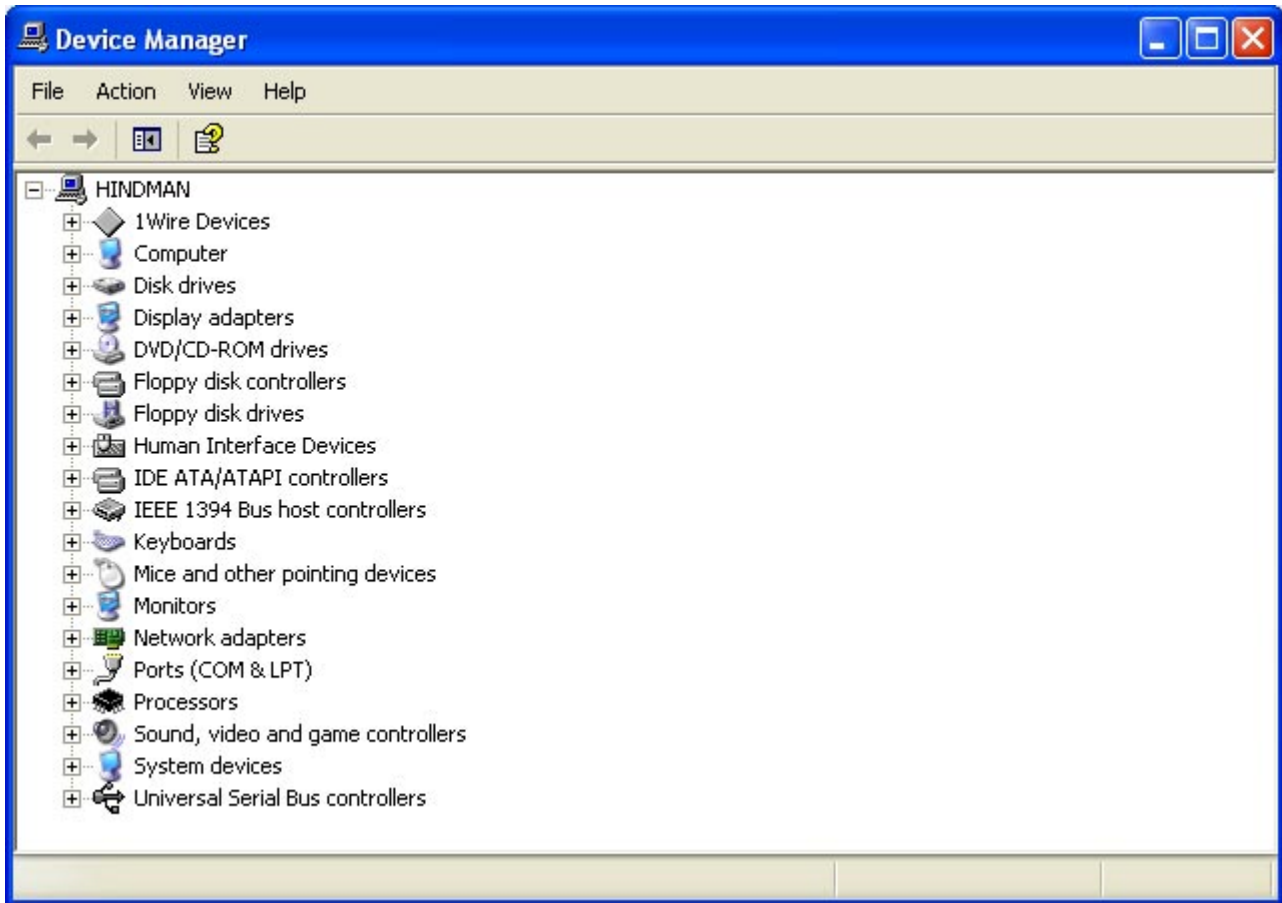


## APPENDIX A: 1-WIRE USB ADAPTER (DS9490) INSTALLATION HELP

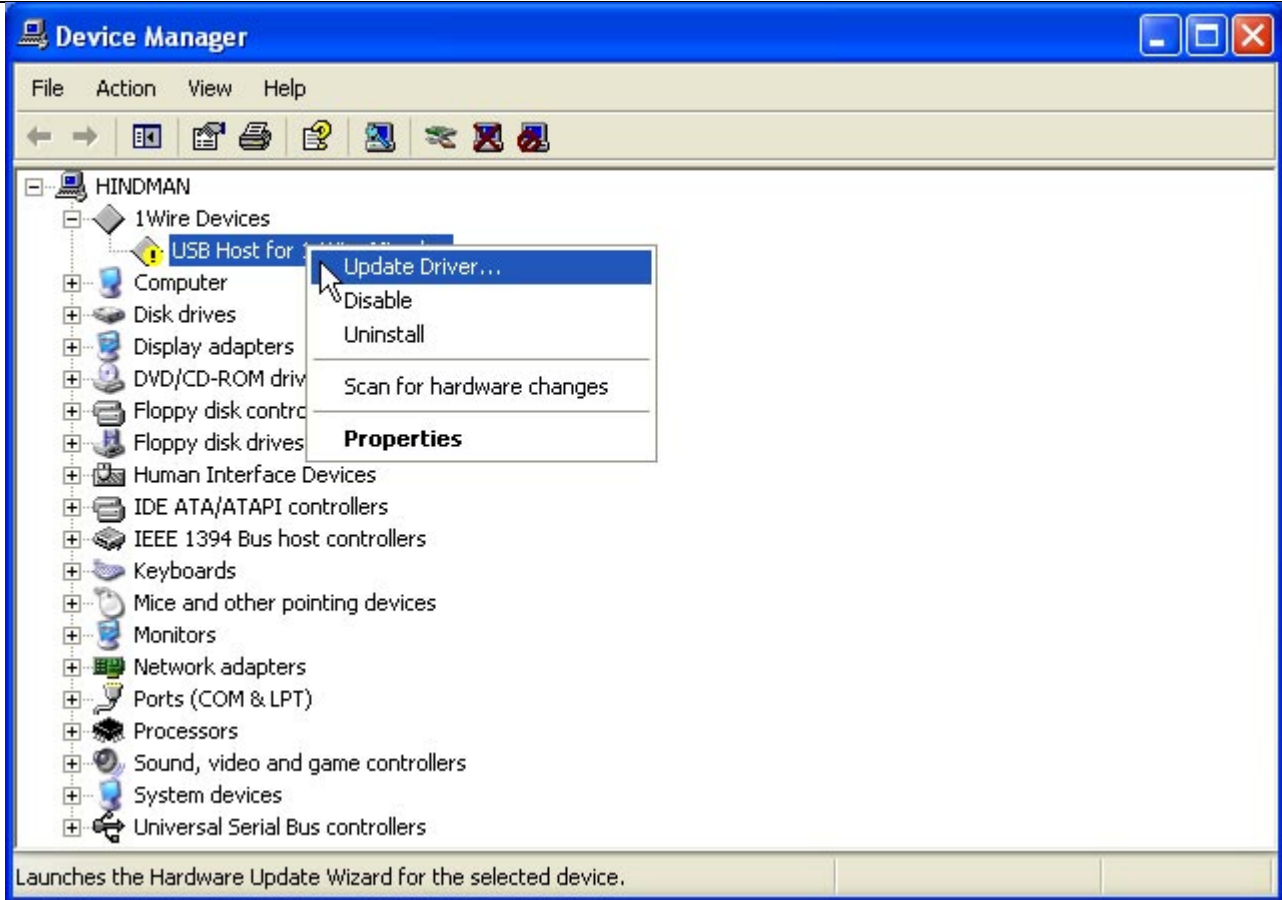
This appendix provides troubleshooting tips for 1-Wire USB adapter installations. Please see the following website for the latest 1-Wire Drivers information and downloadable installation packages:  
<http://www.ibutton.com/software/tmex>.

### Troubleshooting the 1-Wire USB Adapter Device Driver Installation

1. Make sure an up-to-date version of the 1-Wire Drivers has been downloaded. Version 3.XX series of 1-Wire Drivers installs TMEX only. Version 4.00 beta 4 and above installs OWCOM with optional .NET support. Visit <http://www.ibutton.com/software/tmex> to check for the latest versions.
2. If running Windows 2000 or XP, first log into an account with administrator privileges. This is required for installing devices drivers. See note at the end of this document.
3. Uninstall previous 1-Wire Drivers version before installing the above drivers.
4. Attempt to install the drivers mentioned in step 1 by double-clicking on the installation program's icon and following the instructions closely.
5. After installation, if difficulties persist, please look in the "Device Manager" to see if the 1-Wire USB adapter has been installed correctly (it will show up in a list of hardware devices attached to the computer). To do this, from the Control Panel, double-click "System". Then choose the correct tab by clicking on it. In Windows 98, the correct tab is "Device Manager". In Windows XP, the correct tab is "Hardware" (then click on the button labeled "Device Manager").
6. With the USB adapter plugged in, the "Device Manager" should show the following entry, either "1Wire Devices" or "1-Wire Devices". The following is a picture of a good install on Windows XP (with USB adapter plugged in):



7. If the following error appears, indicated below by a yellow circle with an exclamation mark under the "1Wire Devices" entry, then try "Updating" the device driver by right-clicking the entry and then clicking on the "Update Drivers" menu option. (Keep the USB adapter plugged in).



8. Follow the instructions. When prompted for the new driver, browse to the <applications> folder. This is the folder specified when running the installation program above. The driver, DS2490.SYS, and its respective DS2490.INF file should be in the <applications> folder. By default, the <applications> folder is located here: "C:\Program Files\Dallas Semiconductor\1-Wire Drivers Version X.XX". The X.XX is a version number (which can also include "beta" and an extra number). Then proceed to finish the "Update Driver" wizard.
9. The driver should now be correctly installed. See the picture in Step 7 above for a picture of the "Device Manager" with the 1-Wire USB adapter's driver successfully installed.

### Installing the 1-Wire USB Adapter's Device Driver by Hand

1. Two files are necessary to install the 1-Wire USB adapter (DS9490) device driver by hand. The first is the DS2490.SYS file (the actual device driver). And the second is the DS2490.INF file (the file Windows uses to setup the DS2490.SYS file correctly). If an installation of the 1-Wire Drivers has already been attempted, the two files can be found in the <applications> folder. This is the folder specified when running the installation program. By default, the <applications> folder is located here: "C:\Program Files\Dallas Semiconductor\1-Wire Drivers Version X.XX" (with X.XX signifying a version number).
2. Alternatively, download an up-to-date version of the 1-Wire Drivers from the following location: <http://www.ibutton.com/software/tmex>. Extract the files to a specified directory.
3. Make sure the DS2490.INF file and the DS2490.SYS file (the device driver) are present in the directory (these are the only two needed).
4. If running Windows 2000 or XP, first log into an account with administrator privileges. This is required for installing device drivers. See note below on how to determine account privileges.
5. Plug in the DS9490 1-Wire USB adapter.
6. If the "Add Hardware" (or "Add New Hardware") Wizard does not run in response to plugging in the DS9490, then run the Wizard from the Control Panel.
7. Follow the instructions and when asked to browse for the driver. Please browse to the directory containing the DS2490.INF and the DS2490.SYS file.
8. Finish the "Add Hardware" wizard.

\*To determine the account privileges (XP or 2000) follow these steps:

1. Log on to Windows with the user account in question.
2. Click Start > Control Panel.
3. Double-click the User Accounts icon.

If the account is a "Computer Administrator" under XP, icons for all of the accounts available on the computer are shown (including a "Guest" account that is disabled by default). Next to each icon is the user name, and the phrase "Computer Administrator" or "Limited account." If a Limited user, only the icon for the account will be viewable. Next to this icon are the user name, and the phrase "Limited account."

### Simulating a 'First-time' Install

If a developer wants to test new DS9490 drivers, new installation program modifications, or if the initial install failed for any reason and it is necessary to do, in effect, a new 'first-time' driver installation, follow the instructions given below. As to new or alternate drivers, the Public Domain Kit offers a Windows binary build that uses the open source "libusb" device driver instead of the DS2490.SYS. For more information, including installation instructions, please visit the Public Domain Kit's main web page: <http://www.ibutton.com/software/1wire/wirekit.html>. In the un-install instructions below, note that %WIN% stands for the "Windows" directory, which can vary by OS (i.e., C:\Windows, C:\WinNT, etc.). Similarly, %WIN\SYS% stands for the "System" directory, which can also vary (i.e., C:\Windows\System, C:\WinNT\System32, etc.).

#### Instructions:

1. Delete the \*.inf file(s) associated with the DS2490.SYS (including any \*.pnf files). It is important that these files get removed. If this does not happen, Windows tries to install the DS2490.SYS again when the USB adapter gets plugged back in to the PC. Microsoft Windows \*.inf file locations differ between operating systems. See tables below for where to look for these files.
2. Delete the DS2490.SYS file from the %WIN\SYS%\drivers directory.
3. Using the Registry Editor (RegEdit.exe or RegEdt32.exe), purge the registry of the following driver information:

For Windows 98, ME:

Delete the following registry key:

\LocalMachine\System\Enum\USB\Vid\_04fa&Pid\_2490

For Windows 2000, XP:

Delete the following registry key:

\LocalMachine\System\CurrentControlSet\Enum\USB\Vid\_04fa&Pid\_2490

On Windows 2000/XP, you must restart the system before reinstalling the driver.

**Table 5. Inf Files and Locations for Version 3.21 and 4.00 Beta 4 Drivers (and Up)**

| OS              | Inf Files  | Locations  |
|-----------------|--|--|
| Windows XP/2000 | OEMXX.INF<br>(the XX number is determined by the OS) | %WIN%\inf<br>(A few inf files may need to be read until the correct one is discovered) |
|                 | OEMXX.PNF  | %WIN%\inf  |
| Windows ME/98   | OEMXX.INF  | %WIN%\inf  |
|                 | Dallas Semiconductor MAXIMDS2490.INF                 | %WIN%\inf\other  |

**Table 6. Inf Files and Locations for Older 1-Wire Drivers Versions**

| <b>OS</b>     | <b>Inf Files</b>                                     | <b>Locations</b>   |
|---------------|--|--|
| Windows XP    | OEMXX.INF<br>(the XX number is determined by the OS) | %WIN%\inf<br>(A few inf files may need to be read until the correct one is discovered) |
|               | OEMXX.PNF  | %WIN%\inf  |
| Windows2000   | DS24902K.INF   | %WIN%\inf  |
|               | OEMXX.PNF  | %WIN%\inf  |
| Windows ME/98 | DS2490.INF   | %WIN%\inf  |
|               | Dallas Semiconductor Corp.DS2490.INF                 | %WIN%\inf\other  |