QuickUSB Customizer

User Guide

Bitwise systems

Bitwise Systems
6489 Calle Real, Suite E
Goleta, CA 93117
Voice (805) 683-6469
Fax (805) 683-4833
Toll Free (800) 224-1633
Web Site www.bitwisesys.com
Information info@bitwisesys.com
Technical Support support@bitwisesys.com

Version 2.15.2
June 15, 2012

Table of Contents

Overview

The QuickUSB Customizer is a production level tool used to configure QuickUSB hardware to be uniquely identified as a device other than the default "QuickUSB Module" made by "Bitwise Systems". The software customizes information about your hardware such as the USB Vendor ID (VID), Product ID (PID), serial number, and description strings so that when your hardware is connected to a computer it is uniquely identified as a product made and owned by your company. The software can automatically create a driver package for your customized hardware so that the installation of the QuickUSB Driver and QuickUSB DLLs (along with their dependencies) is quick, painless, and tailored around your product. The QuickUSB Customizer may be used on hardware that already contains licensed firmware (QuickUSB Modules, ChipPack EEPROMs, etc.), and is capable of programming blank EEPROMs with firmware using iChipPack licenses. The QuickUSB Customizer also can set the firmware default settings so that production runs of hardware may be fully configured without the need to use any additional software such as the QuickUSB Programmer or QuickUSB Diagnostics Utility.

In order to have a USB device be uniquely identified, every kind of USB device must have a unique Vendor ID (VID) and Product ID (PID) combination. The Vendor IDs are regulated by the USB Implementers Forum (USBIF) and are licensed at up to $2,000 or more. Once a vendor has been licensed a Vendor ID, vendors are free to regulate their Product IDs under their Vendor ID. In order to customize your hardware with the customizer you will need to have your own unique VID and PID combination. Bitwise Systems licenses unique PIDs under their own VID at a cost less than that of a VID as to help dissuade the added cost of licensing a new VID from the USBIF. Individual PIDs may be purchased online from Bitwise Systems at www.quickusb.com and are issued in the form of QUC files. If you already have licensed a VID from the USBIF, then you will need to Email support@quickusb.com to obtain a valid QUC file to enable the enhanced features of the QuickUSB Customizer. Note that each product that must be uniquely identified must have a unique VID/PID combination, but any number of those products may be produced and manufactured without any additional VID/PID licensing.

The QuickUSB Customizer may be used in a reduced feature mode if you do not have or wish not to use a unique VID/PID for your hardware. In this mode you are able to program firmware, set firmware default settings, and set the serial number of the device. In order to change the VID, PID, USB descriptor strings, and generate a custom driver package you will need a valid QUC file that contains the unique VID and PID allotted to you.

System Requirements

- Windows PC (x86 or x64) running XP, Vista, or Windows 7
- QuickUSB Driver v2.15.2
- QuickUSB DLLs v2.15.2
- QuickUSB .NET Assembly v2.15.2
- .NET Framework 3.5 (or later)

Using the QuickUSB Customizer

Creating a Driver Package

Overview

USB devices are identified by their VID/PID combination and associated with a driver that declares it supports USB devices with a matching VID/PID combination. Once the QuickUSB Customizer is used to customize a device with a new VID/PID combination, Windows will be unable to locate a valid driver for the new device. In order for Windows to find and use the QuickUSB Driver with your customized hardware, a new driver package must be created and installed. This driver package is modeled after the QuickUSB Driver Package and is supported on XP, Vista, and Windows 7 (both x86 & x64). Parts of the driver package may be customized to personalize your device on the users system allowing the end-user to interact with (what they perceive as) your custom and personalized hardware instead of a generic QuickUSB Module. Note that in order for the generated driver packaged to be authenticated by Windows (and therefore allowed to be installed), it must be electronically signed. If you are using the QuickUSB Customizer in the reduced feature mode (because you do not have a QUC file) or do not need to generate a driver package, you may skip this section.

Running The Driver Package Wizard

Overview Page

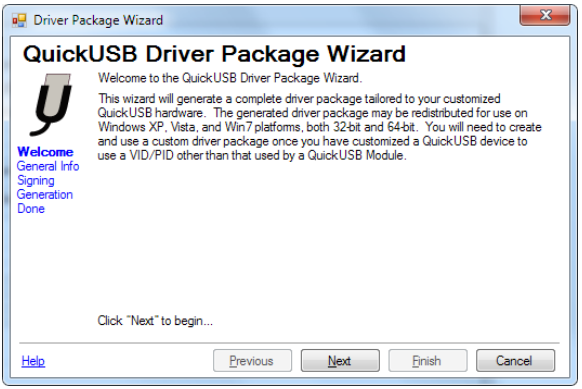

Figure 1: Driver Package Wizard Welcome Page

You may run the QuickUSB Driver Package Wizard after a valid QUC file has been loaded by the software. A QUC file is required because you only need to generate a driver package if you customize a QuickUSB device to use a unique VID/PID. To run the Driver Package Wizard, first open a QUC file, select "Open QUC file..." from the "File" menu and select the QUC file to open. Then, select "Driver Package Wizard..." from the "Tools" menu. The first page of the wizard is a welcome screen that summaries the use of the wizard and supported platforms. Currently the Driver Package Wizard supports 32-bit and 64-bit versions of Windows XP, Vista, and Win7. Click the "Next" button to continue with the wizard.

General Information Page

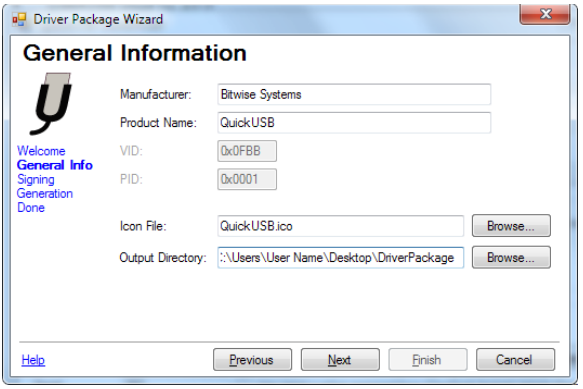

Figure 2: Driver Package Wizard General Information Page

The General Information Page of the wizard is where you can specify information specific to your product. This information includes the manufacturer, product name, icon (shown in the driver installer and in the Add/Remove Programs editor), and package output directory. Note that if the output directory already exists you will later be prompted to overwrite the contents of that existing directory. The VID and PID are locked fields that cannot be altered. Those files are extracted from the QUC file that was loaded before the wizard was executed. Once all the required information is filled out, click "Next" to continue with the wizard.

Driver Signing Page

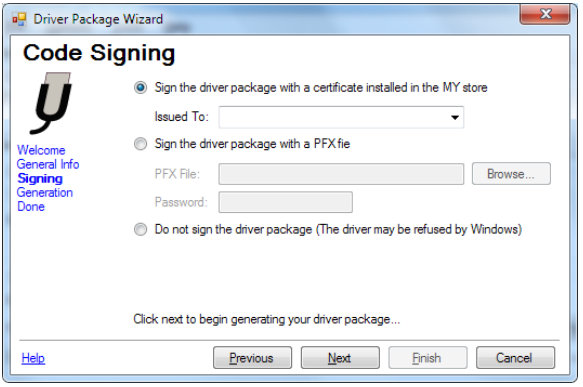

Figure 3: Driver Package Wizard Code Signing Page

All drivers must be electronically signed (code signed) in order to be authenticated and accepted as valid driver's by Windows. The QuickUSB Module driver is signed by Bitwise Systems. When that driver package is executed, Windows requires elevated rights to install the driver and first prompts the use for permission to install the software published/signed by Bitwise Systems, as shown in Figure 4. In a driver is not signed (Figure 5), Windows still prompts the user for elevated rights and perform the first step of driver installation, but when a device is connected to the system that requires the kernel to load that driver the load fails because the driver is not trusted software and reports the device as not working properly. In order to use the driver generated by the Driver

Package Wizard, you must sign the driver with a code-signing certificate issued to you and installed in the Certificate Store on the computer running the Driver Package Wizard. The wizard offers two ways for you to automatically perform the code signing for you: (1) You may enter the "Issued To" name of the certificate in the "MY" Certificate Store, or (2) You may direct the wizard to the PFX file to use to perform the code signing. The "Issued To" combo box is populate with the names of certificates found in the "MY" Certificate Store. You may select one from the drop-down list or manually type in the certificate name. When using PFX files, the PFX file must contain the private key and, if the file is password protected, you must specify the password. If you opt to not have the wizard perform the code signing step, then you will later need to manually sign the catalog file (with the ".cat" extension) and the "Setup.exe", "DPInstx86.exe", and "DPInstx64" executable files. Once you have selected your preferred signing method, click the "Next" button to begin generating the driver package.

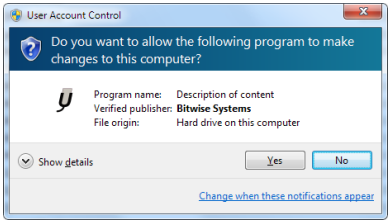

Figure 4: UAC Dialog – Signed (Win7)

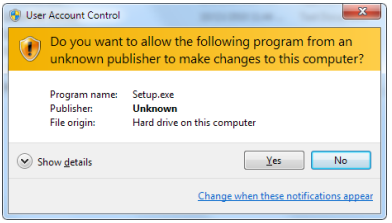

Figure 5: UAC Dialog – Not signed (Win7)

Package Generation Page

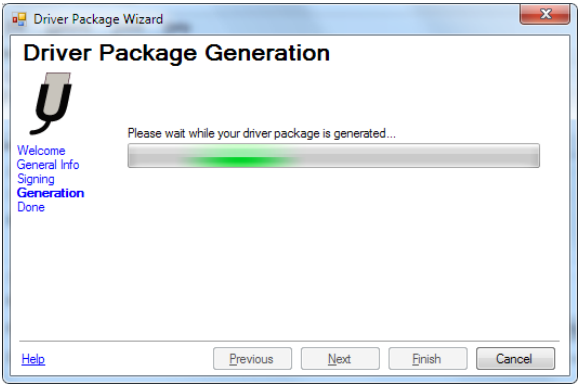

Figure 6: Driver Package Wizard Generation Page

Once you reach this page, the wizard starts generating your driver package. Please do not interrupt this process as doing so may cause a broken/invalid driver to be generated. This process typically takes under a minute to complete, depending on your system. Note that to successfully perform the driver signing step, your computer must be connected to the internet.

Successful Package Generation Page

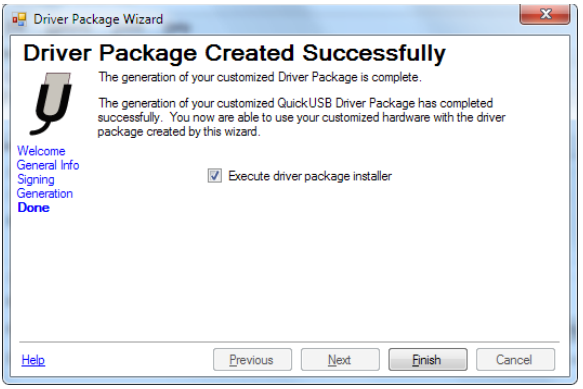

Figure 7: Driver Package Wizard Success Page

At this stage your driver package has been successfully generated and placed in the output directory specified in the General Information page. You have the option to execute the installer to begin installation on your computer. Note that if you choose to execute and install the driver from the wizard, Windows may prompt you to reboot your computer after installing the driver since the driver package installs the QuickUSB DLLs, which are currently in use by the QuickUSB Customizer application.

Unsuccessful Package Generation Page

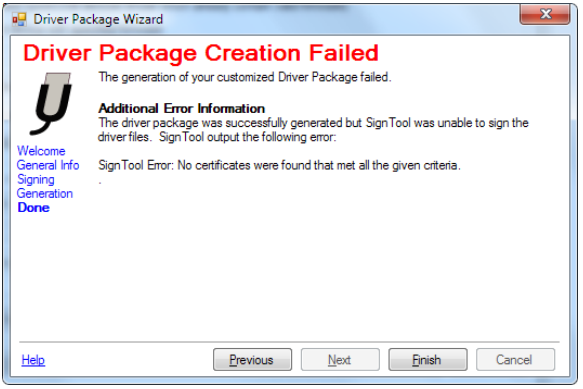

Figure 8: Driver Package Wizard Failure Page

If the creation of the driver package failed, you will be directed to an error page containing additional information about the failure. You may use this information to help correct the error and make another attempt at generating the driver package. If still are unable to successfully generate a driver package or unable to understand the error being reported please contact the QuickUSB Support team at support@quickusb.com.

Uninstalling the Driver

After the driver has been successfully installed, a new item is listed in the Windows Add/Remove Programs Manager. The item will be called “Windows Driver Package – [ManufacturerName] [ProductName] (DateStamp Version)”, where “ManfacturerName” and “ProductName” are the respective strings specified on the “General Information”

page of the driver wizard, "DateStamp" is the date/time stamp of the driver, and "Version" is the version of the driver. The icon next to the driver package item in the programs list is the one specified on the "General Information" page of the driver wizard. The user may remove the installed driver at any time by opening the Add/Remove Programs Manager in Windows, navigating to the driver package item, and choosing to uninstall the driver.

Customizing Licensed Devices

Overview

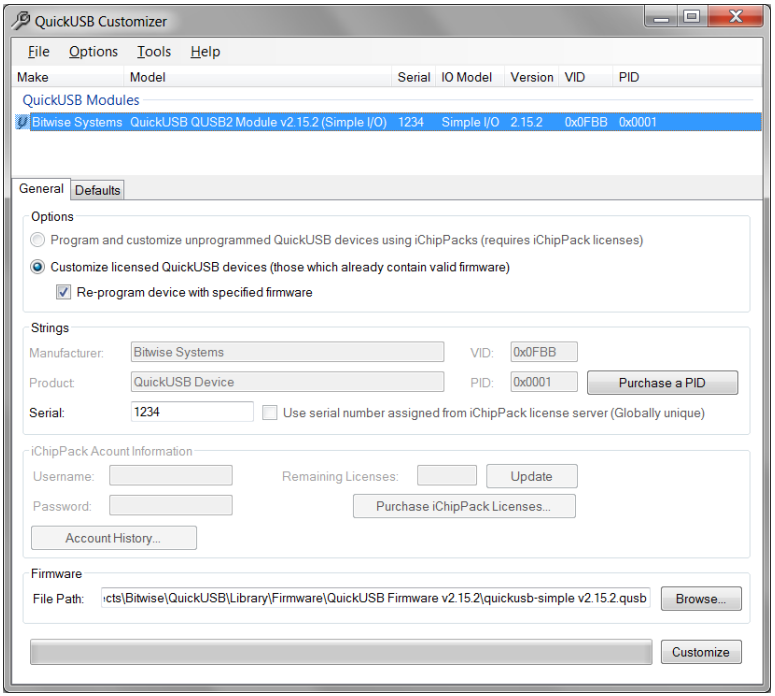

Figure 9: QuickUSB Customizer with Licensed Firmware

When a QuickUSB device is connected to the computer, it will show up in the "QuickUSB Modules" control list of the QuickUSB Customizer. The customizer will automatically determine if the hardware is licensed (i.e. if the EEPROM already contains valid firmware) and select the appropriate customizing method. If a licensed device (such as a QuickUSB Module, hardware build with ChipPack EEPROMs, hardware build with iChipPack licenses that have already been programmed, etc.) is detected, the "Customize licensed QuickUSB devices (those which already contain valid firmware)" will be selected. When that option is selected, all of the items underneath the "iChipPack Account Information" become disabled, as they are not used for this process.

Customizable Strings

The QuickUSB firmware has three strings that may be customized by the QuickUSB Customizer: the manufacturer name, the product name, and the serial number. These

strings are returned by the QuickUsbGetStringDescriptor() API call and are used by Windows and the USB system to identify your device hardware.

Note that the manufacturer and product name string may only be altered after a QUC file has been loaded. Without a QUC file loaded, the VID and PID of the device are locked to 0x0FBB and 0x0001, respectively, which define the device to enumerate as a QuickUSB Module. Also, note that the "Manufacturer" string is limited to 31 Unicode characters, the "Product Name" is limited to 49 Unicode characters, and the serial number is limited to 8 Unicode characters (the serial number is stored as a string in firmware and therefore does not have to be limited to only numeric values). To open a QUC file, select "Open QUC file..." from the "File" menu and select the QUC file to open. After a valid QUC file has been opened, the VID/PID values are updated and the manufacturer name and Product name text boxes become editable.

Firmware Programming

When customizing a device that already contains valid firmware, you have the option to reprogram the device with another firmware file. This process programming process is the same as using the QuickUSB Programmer but has the added convenience of allowing you to not have to use a second piece of software to fully configure and customize a QuickUSB device. The firmware textbox and browse buttons are enabled when the "Re-program device with specified firmware" option is selected in the "Options" group box.

Default Settings

The QuickUSB default settings may be set for the QuickUSB device by navigating to the "Defaults" tab and specifying the appropriate defaults. Specifying device defaults is an optional process and should only be done when required, as incorrect settings may lead to improper device behavior. To ensure that the device defaults are set properly, it is best to first read the defaults from the QuickUSB by selecting "Read defaults from selected device" in the "Options" menu, then altering the defaults as required. Default settings are stored in QUC files so that you may customize QuickUSB devices without having to manually select the proper defaults for each device. To save changes to the default settings to an opened QUC file, select "Save changes to QUC file" from the "File" menu.

Customizing Unlicensed Hardware with iChipPack Licenses

Overview

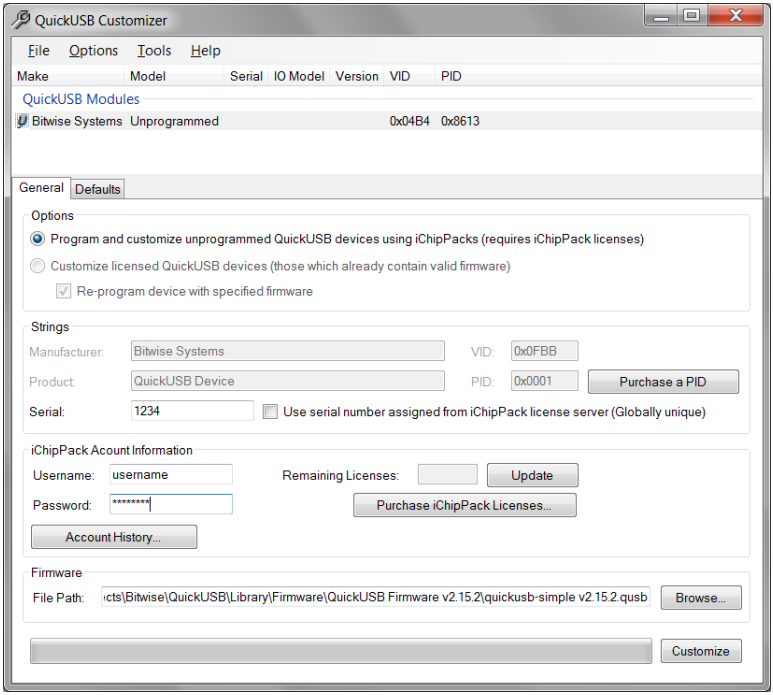

Figure 10: QuickUSB Customizer with no Firmware

When a QuickUSB device is connected to the computer, it will show up in the "QuickUSB Modules" control list of the QuickUSB Customizer. The customizer will automatically determine if the hardware is licensed (i.e. if the EEPROM already contains valid firmware) and select the appropriate customizing method. If a non-licensed device (such as hardware build with blank EEPROMs) is detected, the "Program and customize unprogrammed QuickUSB devices using iChipPacks (requires iChipPack licenses)" will be selected. When that option is selected, all of the items underneath the "iChipPack Account Information" become enabled, as they are required for this process.

Customizable Strings

The QuickUSB firmware has three strings that may be customized by the QuickUSB Customizer: the manufacturer name, the product name, and the serial number. These strings are returned by the QuickUsbGetStringDescriptor() API call and are used by Windows and the USB system to identify your device hardware.

Note that the manufacturer and product name string may only be altered after a QUC file has been loaded. Without a QUC file loaded, the VID and PID of the device are locked to 0x0FBB and 0x0001, respectively, which define the device to enumerate as a QuickUSB Module. Also, note that the "Manufacturer" string is limited to 31 Unicode characters, the "Product Name" is limited to 49 Unicode characters, and the serial number is limited

to 8 Unicode characters (the serial number is stored as a string in firmware and therefore does not have to be limited to only numeric values).

iChipPack Account Information

To use iChipPack licenses, you will need to specify your iChipPack account name and password. This information is provided to you when you first purchase iChipPack licenses. If you need to purchase iChipPack licenses for the first time, or would like to purchase additional iChipPack licenses, click on the "Purchase iChipPack Licenses..." button, which will open a web browser to the appropriate page on the QuickUSB website. You may also use the "Account History..." button to get access to your iChipPack account history online. To check the number of remaining licenses you have in your account, clock on the "Update" button to the right of the "Remaining Licenses" label.

Firmware Programming

Devices that are being configured and customized with iChipPack licenses do not already contains valid firmware so you must specify the firmware file you would like for the device. The programming process used here is the same as using the QuickUSB Programmer but has the added convenience of allowing you to not have to use a second piece of software to fully configure and customize a QuickUSB device.

Default Settings

The QuickUSB default settings may be set for the QuickUSB device by navigating to the "Defaults" tab and specifying the appropriate defaults. Specifying device defaults is an optional process and should only be done when required, as incorrect settings may lead to improper device behavior. To ensure that the device defaults are set properly, it is best to first read the defaults from a known-good QuickUSB device. This may be done by selecting "Read defaults from selected device" in the "Options" menu, then altering the defaults as required. Default settings are stored in QUC files so that you may customize QuickUSB devices without having to manually select the proper defaults for each device. To save changes to the default settings to an opened QUC file, select "Save changes to QUC file" from the "File" menu.