



# USB Device Application Kit for Windows Embedded CE 6.0

## User's Manual

Digi document reference number: 90000926\_A

© Digi International Inc. 2008. All Rights Reserved.

The Digi logo is a registered trademark of Digi International, Inc.

All other trademarks mentioned in this document are the property of their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Digi International Inc.

11001 Bren Road East

Minnetonka, MN 55343 (USA)

+1 877 912-3444 or +1 952 912-3444

<http://www.digiembedded.com>

# Contents

1	Introduction.....	6
<b>1.1</b>	<b>Features.....</b>	<b>6</b>
2	Requirements.....	7
<b>2.1</b>	<b>BSP requirements.....</b>	<b>7</b>
<b>2.2</b>	<b>Microsoft Windows® CE OS Configuration .....</b>	<b>7</b>
3	Installation .....	9
4	Integration.....	10
<b>4.1</b>	<b>Integration overview .....</b>	<b>10</b>
<b>4.2</b>	<b>Integration process.....</b>	<b>10</b>
<b>4.3</b>	<b>Driver starts.....</b>	<b>12</b>
5	Using the USB Device functionality .....	13
<b>5.1</b>	<b>Serial USB Function Client .....</b>	<b>13</b>
<b>5.2</b>	<b>Mass Storage USB Function Client.....</b>	<b>17</b>
6	Uninstallation.....	18



## Conventions used in this manual

Here is a list of the typographical conventions used in this manual:

<b>Style</b>	New terms and variables in commands, code, and other input.
<i>style</i>	In examples, to show the contents of files, the output from commands. In, text the C code.  Variables to be replaced with actual values are shown in italics.
<b>Style</b>	For menu items, dialogs, tabs, buttons, and other controls.  In examples, to show the text that you enter
Menu name > option	A menu followed by one or more options; for example, File > New.

This manual also uses these frames and symbols:



---

**This is a warning. It helps solve or avoid common mistakes or problems.**

---



---

*This is a hint. It contains useful information about a topic.*

---



```
> This is a host computer session  
> Bold text indicates what must be entered.
```



```
> This is a target session  
> Bold text indicates what must be entered.
```



## Abbreviations

ASCII	American Standard Code for Information Interchange
CPU	Central Processing Unit
ESD	Electrostatic Discharge
GPIO	General Purpose Input/Output
OS	Operating System
PC	Personal Computer
RAM	Random Access Memory
ROOTFS	Root File System
TFTP	Trivial File Transfer Protocol
TTY	Teletypewriter
USB	Universal Serial Bus



# 1 Introduction

The USB Device Application Kit is a product that brings the USB Device physical interface to your Digi Development board or JumpStart board.

The USB Device Application Kit hardware board connects to the Development or JumpStart board via the USB Application Header and the provided 16-pin flat cable. It basically mounts a USB PHY and USB B female connector.

The USB Device Application Kit software contains the driver needed to support the USB Device Controller in a Microsoft Windows® Embedded CE 6.0 OS Design.

This User's Manual assumes that the reader is able to create, compile and download a kernel image.

## 1.1 Features

---

The following are the main features of the USB Device Application Kit:

- USB 2.0 Full Speed (Low speed not supported)
- Board is powered from Development or JumpStart board
- Operation modes: serial and mass storage,
- $\pm 15\text{kV}$  ESD protection



## 2 Requirements

### Host System requirements

The development system has to meet the following requirements:

- x86 PC with 500 MHz Pentium III or faster processor; 2 GHz Pentium 4 or equivalent recommended
- Microsoft Windows® 2000 Professional with Service Pack 4 or Windows XP Professional with Service Pack 1.
- Microsoft Windows® CE 6.0 with Service Pack 1 installed and updates
- Serial port
- Ethernet network card

### 2.1 BSP requirements

---

Display interface support recommended for development purposes, but not required.

### 2.2 Microsoft Windows® CE OS Configuration

---

#### 2.2.1 Required components

Go to the Catalog and expand **Device Drivers > USB Function > USB Function Clients**. Then include this element:

**USB Function Clients:**

- Mass Storage
- serial

When using ActiveSync the components from **Core OS > CEBASE > Applications – End User > ActiveSync** must be included into the Windows CE project.

Then include this element:

**ActiveSync:**

- File Sync

Furthermore, the component from **Core OS > CEBASE > Communication Services and Networking > Networking – Wide area Network (WAN)** must be included into the Windows CE project

Then include this element:

**Telephony API (TAPI 2.0):**

- Unimodem Support



## 2.2.2 Recommended catalog components

Other recommended networking utilities and services are:

### **Core OS > CEBASE > Communication Services and Networking:**

- Networking General
  - Network Utilities
- Servers
  - FTP Server
  - Telnet Server





## 3 Installation

This release is installed by executing Setup.exe.

The installer wizard will guide you in all required steps.

After installation has finished, the following components will be on your PC:

*%ProgramFiles%\Digi\AppKits\USB\_Device\_AppKit:*

- **Uninstaller:** Executable to uninstall this release.
- **Release Notes and License Agreements.**

*%\_WINCEROOT%\OTHERS\Digi\AppKits\USBFN\src\driver:*

- Sources of the driver.



---

*%ProgramFiles% is an environment variable of your system that provides the path to your Program Files directory (usually C:\Program Files)*

*%\_WINCEROOT% is an environment variable of your system that provides the path to your Microsoft Windows® CE root directory (usually C:\WINCE600).*

---

## 4 Integration

### 4.1 Integration overview

---

The USB Device Application Kit integration described in this chapter assumes that you have already performed the following steps. These steps are fairly general as they are highly dependant on the customer BSP and the Microsoft Windows® CE version used:

Before completing these steps, you will need to have the following applications already installed:

- Windows Embedded as described in Chapter 2, and
- a BSP corresponding with the hardware that will be used.

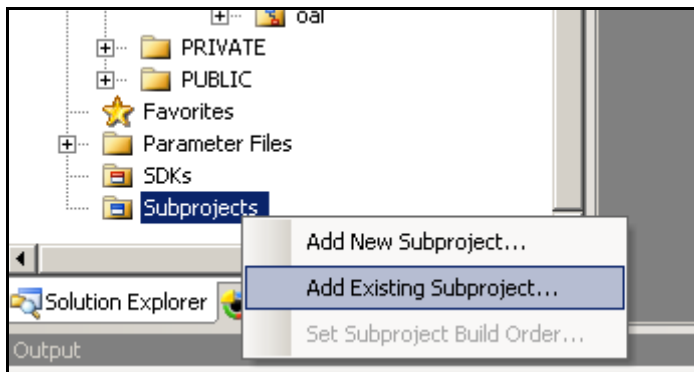
### 4.2 Integration process

---

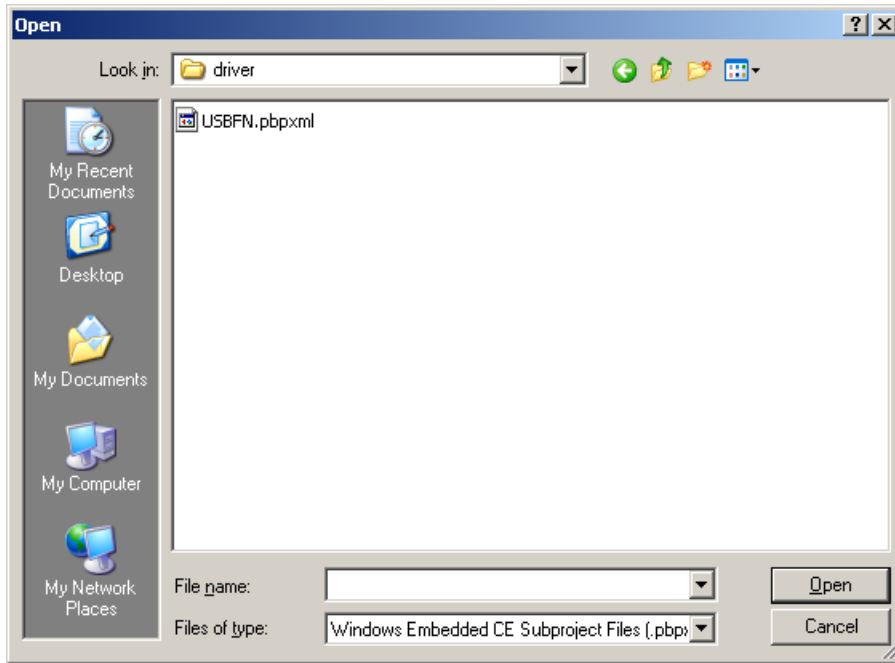
The source code of the driver is also available and can be integrated into a Windows Embedded CE 6.0 project as a subproject of it.

After you have completely built your Project, it's time to add a new subproject for the USB\_Device\_AppKit Module.

Open the Solution Explorer and right-click over the Subprojects. Choose **Add Existing Subproject**:



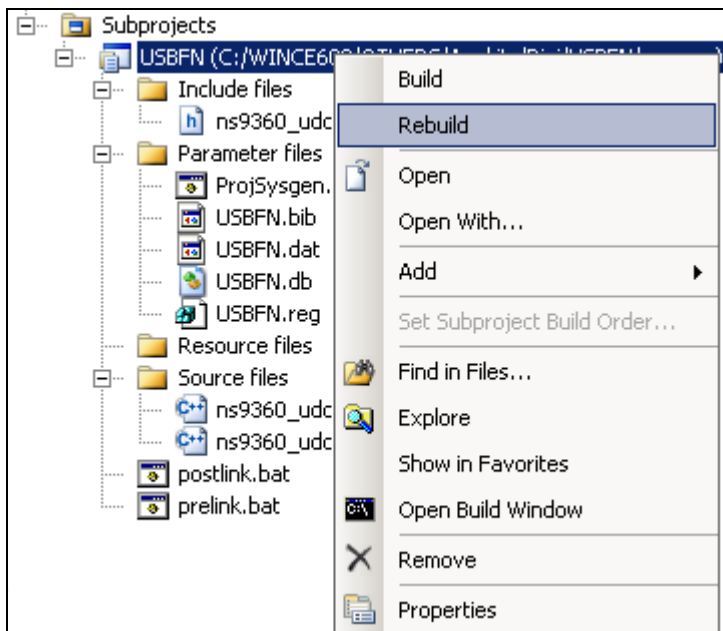
Navigate to `%_WINCEROOT%\OTHERS\Digi\AppKits\USBFN\src\driver` directory and select USBFN.pbxml:



Now you can see the parameter files and the source files.

Right click over USBFN and select **Rebuild** and then right click over the project and select **Make Run-Time Image**.

The driver build should be rebuilt and included in the final **nk.bin** image, together with the necessary registry entries.





## 4.3 Driver starts

---

Verify that the **ns9360\_usbfn.dll** is inside the new image and that the correct registry entries are placed in the final registry file (reginit.ini).

Once the project has been built, download it to the target device and start it up.

If you have the USB Device Application Kit connected to the peripheral application header of the target, it should be automatically detected, and you should see a console message similar to the following:



```
[UfnPdd_Init]: Loading USB Device Driver... OK.
```



---

*More information may be shown depending on the debug level established in the driver.*

---



---

*In the console an error message will show up telling that an IOCTL isn't supported by the driver. That message can be ignored because that IOCTL only receives some interface information that are not important to the functionality of the driver*

---



## 5 Using the USB Device functionality

The USB Device driver provides the functionality upon which the USB function client drivers work. The USB function client drivers implement one or more "functions", each providing a different capability to the USB host. In this document we'll see two of these function client drivers:

- Serial USB Function Client
- Mass Storage USB Function Client

### 5.1 Serial USB Function Client

---

For both serial profiles standard serial support is necessary within the Windows CE image. That component can be found at **Core OS > CEBASE > Core OS Services > Serial Port Support**.

#### 5.1.1 Selecting the ActiveSync profile

Make the following change in your **USBFN.reg** file, you need to uncomment the **Serial\_Class** and comment the rest (";").



```
[HKEY_LOCAL_MACHINE\Drivers\USB\FunctionDrivers]
; "DefaultClientDriver"="Mass_Storage_Class"
"DefaultClientDriver"="Serial_Class"
; "DefaultClientDriver"="USBSER_Class"
```

Now you can build the project with the profile selected.

Right click over USB\_Device\_AppKit and select **Rebuild** and then right click over the project and select **Make Run-Time Image**.

#### 5.1.2 Using the profile for ActiveSync

On the PC side (Host) ActiveSync 4.x or higher must have been installed previously.



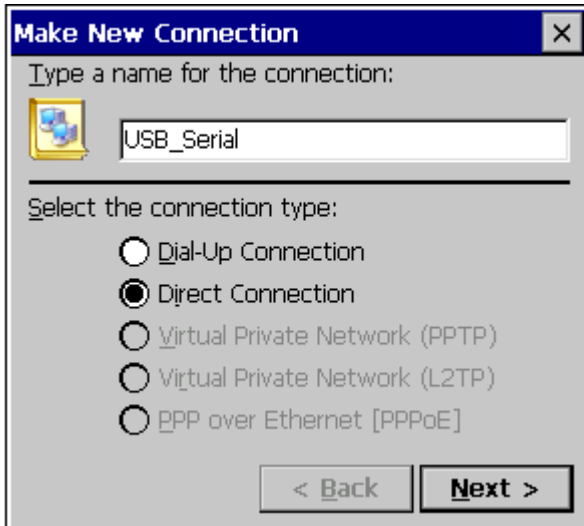

---

*To use ActiveSync some additional components are necessary. Refer to chapter 2.2.1 and include the listed components. If the components have been included after building the project a SYSGEN must be done again.*

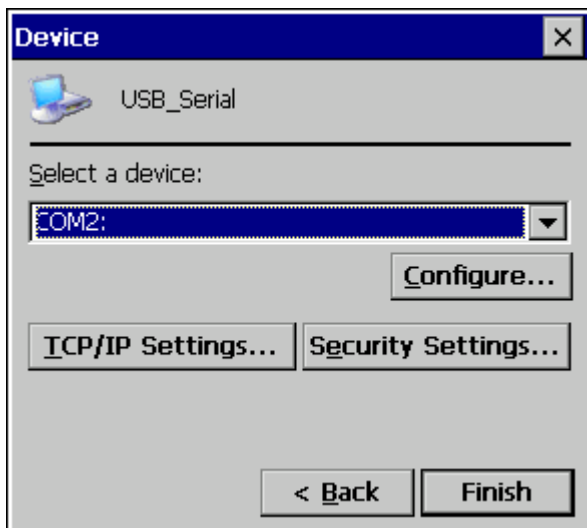
---

After building and deploying successfully the Windows CE image a ActiveSync connection might be necessary to be created on the target side. This is only necessary if more than one serial port is enabled on the target device.

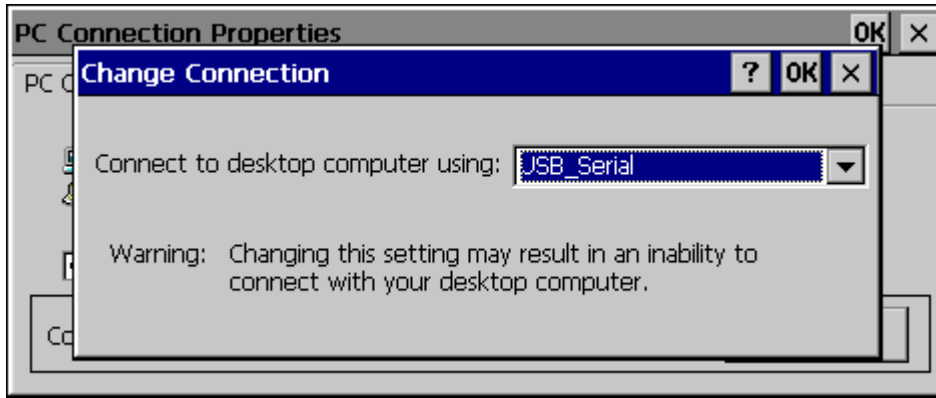
Following the steps that are necessary to create on the target side a new direct connection that can be used later by ActiveSync.



From the *Start* menu select *Settings -> Network and Dial-up Connections* and create a new *Direct Connection* selecting the *Make New Connection* option. Select on the next page the serial port that was setup by the system for the USB function interface and close the dialog. The information on the following picture is only an example and that value might be different on other platforms.



Now the new connection to use the USB function driver is created. The next step is to use that connection as the default configuration for ActiveSync. Select from the *Start* menu *Settings -> Control Panel* and open the *PC connection* item.



Click on the Change button and select the new created direct connection from the list of available connections. Confirm both dialogs with *OK* and now when connecting the USB function device to the PC ActiveSync should automatically start.

### 5.1.3 Selecting the Serial profile

Make the following change in your **USBFN.reg** file, you need to uncomment the **Serial\_Class** and comment the rest (“;”).



```
[HKEY_LOCAL_MACHINE\Drivers\USB\FunctionDrivers]
; "DefaultClientDriver"="Mass_Storage_Class"
; "DefaultClientDriver"="Serial_Class"
"DefaultClientDriver"="USBSER_Class"
```

Now you can build the project with the profile selected.

Right click over **USB\_Device\_AppKit** and select **Rebuild** and then right click over the project and select **Make Run-Time Image**.

### 5.1.4 Using the Serial profile

When using the serial profile no additional components are necessary to expose a serial port to the Host.

On the PC side (Host) the standard USB serial driver (*usbser.sys*) need to be installed. After building and deploying successfully the Windows CE image connect the USB cable with the device connector to the target and with the host connector to a PC or Laptop. The first time the host will install the corresponding driver and the device will be available for use. In the Device Manager the COM number assigned to the serial port can be found and normal serial communication can be used between the target and the connected PC.

### 5.1.5 Installing the USB Serial driver under Windows

In case the USB serial driver isn't installed on the used Windows machine the *usbser.sys* file comes with various versions of Windows. It can be found on Windows XP typically in

```
C:\WINDOWS\Driver Cache\i386\driver.cab
```

When the generic serial driver is loaded and the USB device connected to the Windows host with a USB cable, Windows should recognize the generic serial device and ask for a driver. Indicate Windows to find the driver in the folder that contains *usbser.inf* and *usbser.sys*.

The *usbser.inf* is included in the Kit as an example and can be found at

```
C:\Program Files\Digi\AppKits\USB_Device_AppKit\inf
```



On Windows XP, when the generic serial device is first plugged in, the "Found New Hardware Wizard" starts up. Select "Install from a list or specific location (Advanced)", on the next screen select "Include this location in the search" and enter the path or browse to the folder containing *usbser.inf* and *usbser.sys*.

Windows will complain that the Generic Serial driver has not passed Windows Logo testing, but select "Continue anyway" and finish the driver installation. In the "Device Manager" (under "Control Panel", "System", "Hardware") on Windows XP expand the "Ports (COM & LPT)" entry and you should see *Generic Serial Port* listed as the driver for one of the COM ports.

#### 5.1.6 Uninstalling the USB Serial driver under Windows

To uninstall the Windows XP driver for *Generic Serial Port*, right click on the *Generic Serial Port* entry in the "Device Manager" and select "Uninstall".





## 5.2 Mass Storage USB Function Client

---

### 5.2.1 Selecting the profile

Make the following change in your **USBFN.reg** file that is part of the subproject, you need to uncomment the **Mass\_Storage\_Class** and comment the rest (“;”).



```
[HKEY_LOCAL_MACHINE\Drivers\USB\FunctionDrivers]
"DefaultClientDriver"="Mass_Storage_Class"
;"DefaultClientDriver"="Serial_Class"
;"DefaultClientDriver"="USBSEK_Class"
```

Now you can build the project with the profile selected.

Right click over USB\_Device\_AppKit and select **Rebuild** and then right click over the project and select **Make Run-Time Image**.

### 5.2.2 Using the profile

To use the Mass Storage profile a block device like CF, SD card or USB Memory stick is necessary to make the profile working correctly. Following some examples how some of these block devices need to be added to the Windows CE project.

Exposing a USB memory stick from the target to a PC (Host) the components USB Mass Storage Function client, USB Host support and USB Storage class driver are necessary. Using any other



---

*When connecting the USB cable to the Host it might take some time till the storage device is represented.*

---



## 6 Uninstallation

The USB Device Application Kit can be uninstalled separately using Windows Control Panel.

**Add or Remove Programs:**

- **USB Device Application Kit for Windows Embedded CE 6.0**