Device Emulator
Device Emulator

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

The Device Emulator emulates the behavior of a Windows CE- or Windows Mobile-based hardware platform.

In This Section

About the Device Emulator

Describes the more important features of the Device Emulator.

Device Emulator System Requirements

Lists requirements and recommendations for system memory, processor type, free disk space, and required version of ActiveSync.

Device Emulator Manager

Summarizes the uses of this tool that provides the logical functionality of a physical cradle.

Saved-State Files

Describes how to use this feature to save the complete state of an emulator image.

Platform Manager and Emulator Communication

Describes this technology for remote tool connectivity in a Platform Builder environment.

Keyboards

Provides information about the relationship between physical keyboards and keypresses received by the emulator.

Skins

Summarizes the notion of skins and how to customize them.

Troubleshooting Connection Issues

Presents various connection scenarios and symptoms and the steps to correct them.

Configuring and Using the Device Emulator

Presents a series of common emulator tasks, and the steps to implement them.

Device Emulator Reference

Contains reference material for command-line launch, an XML schema for customizing skins, key combinations, and user interface components.

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

About the Device Emulator

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

The Microsoft Device Emulator version 1.0 is a desktop application that emulates the behavior of a Windows CE- or Windows Mobile-based hardware platform. Using the Device Emulator, you can run, test, and debug a run-time image without the need for a physical device.

Features

A number of features of the Device Emulator significantly improve on its predecessor. The Device Emulator:

- Runs code compiled for ARM processors rather than x86 processors. In most cases, you can run the same binaries on the emulator as you do on the device.
- Supports synchronizing with ActiveSync. You can use the emulator with a full ActiveSync partnership. With this feature you can debug applications that are synchronizing, or use real synchronized data from within the emulator.
- Supports more development environments, including Visual Studio 2005, Visual Studio .NET 2003, and embedded Visual C++ 4.0 (Service Pack 4)—all using ActiveSync.
- Supports GAPI. You can write and debug games on this emulator.

Adaptability

You can configure this virtual hardware platform just as you would a real hardware platform. You can specify screen resolution and orientation, memory size, skin design, and other properties. You can write event-handling code for the simulated hardware buttons and soft keys. The major limitation is that you cannot simulate performance, because the performance of the emulator depends largely on the speed of the processor of your development computer, the amount of system memory available, and other factors that would not affect performance on a distinct physical device. For more information, see Configuring and Using the Device Emulator.

Saving State

You can run one or more instances of the Device Emulator, and you can save the state of any instance for later development. For more information, see Saved-State Files.

Skins

You can apply existing customized skins or write your own skin files. For more information, see Skins.

Launching

How you launch the Device Emulator depends on your installation. Device Emulator 1.0 is available as a download from the Web and as part of the Visual Studio box. For more information, see How to: Launch the Device Emulator.

See Also

Other Resources

Device Emulator

Device Emulator

Device Emulator System Requirements

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The development computer must meet the requirements in the following table to run one instance of the Device Emulator configured to use 64 megabytes of memory.

Item	Minimum	Recommended
System memory	256MB	512
Processor	600 MHz Pentium	1.8 GHz Pentium
Free disk space	50 MB	50 MB
Microsoft ActiveSync	Version 3.7.1 or later	Version 3.7.1 or later

Each additional instance requires additional system memory that is the total of:

- 32 MB; and
- The amount the instance is configured to use.

For example, a second instance configured to use 64 megabytes of memory requires 32 + 64 MB; that is, 96 MB of system memory on the development computer.

See Also

Other Resources

Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Device Emulator Manager

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

Use the Device Emulator Manager (DEM) to start, connect to, and manage Device Emulator instances on your development computer. Using the DEM menu commands, you can connect to the emulator, cradle and uncradle it, shut it down, reset it, bring it to the front, clear saved states, refresh images, and perform other tasks.

The Device Emulator Manager appears as a window showing a tree view of emulators installed on the development computer.

The following paragraphs describe the more important functionality that the DEM provides.

Launching the Device Emulator Manager

Launching the Device Emulator Manager differs among different installations. For more information, see How to: Launch the Device Emulator Manager.

Support For ActiveSync Communication

The DEM facilitates communication to the Device Emulator over ActiveSync by providing the same logical functionality as a physical device cradle. When a device is cradled, it can communicate using ActiveSync, synchronizing contents of email, calendar, application installations, and so on. The emulator by itself has no equivalent functionality and cannot work with ActiveSync unless it is cradled. Using the emulator with the DEM and ActiveSync is analogous to having a physical device and a cradle. A common scenario is to launch the Device Emulator, cradle it using the DEM, and then establish an ActiveSync partnership with it. For more information, see How to: Cradle and Uncradle the Device Emulator.

Support For Standalone Installations of the Device Emulator

In a standalone installation, the datastore that maintains images and information for the emulator is not present. A typical scenario is when you download the Device Emulator onto a development computer where Visual Studio 2005 is not installed.

In such a scenario, the DEM provides a graphical interface for you to select an OS image for the emulator (click **Restore Image** on the **File** menu) and to perform other management tasks that you would normally do only in an integrated development environment, such as Visual Studio.

Additional Functionality

You can also perform the following tasks using the DEM window:

- You can right-click an emulator version in the tree, such as Pocket PC 2003 SE Emulator, and launch that version from the shortcut menu.
- You can retrieve a previous image by clicking **Restore Image** on the **File** menu.
- You can obtain the status of all emulators, whether running, closed, or cradled, by clicking **Refresh**.
- Right-clicking any emulator version in the tree to open the shortcut menu, you can shut down the emulator, reset it (soft), and clear a saved state.
- You can disable prompts to enable network connectivity, serial ports, and folder sharing when a saved-state file uses these features. Click **No Security Prompts** on the **File** menu.

  Security Note
  Any use of folder-sharing poses a potential security risk. Use caution in disabling these prompts.

See Also

Other Resources

Device Emulator

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

A file with the extension .dess (Device Emulator Saved State) contains all RAM, ROM, and settings information from the Device Emulator. The .dess file is therefore the only file necessary to restore a saved state. The .dess extension is registered at setup time, so that double-clicking a .dess file launches the Device Emulator with the state saved in that file.

Security Note
Using saved-state files from untrusted sources can pose a security risk.

Saving State

You can save the state of the Device Emulator at any time by clicking **Save State** on the emulator **File** menu. You are also prompted to save state whenever you close the emulator.

The saved-state file is located by default under \Documents and Settings\<username>\Application

Data\Microsoft\Device Emulator. Global saved-state files are located under \Documents and Settings\All Users\Application Data\Microsoft\Device Emulator.

Note
Saved-state images are created by default on a per/user basis. If you want to create a global saved-state image, launch the emulator from the command line using the /g switch. For more information, see Device Emulator Command Line Reference.

State-saving options in the user interface are as follows:

Prompt	Action
Close without saving	Closes the Device Emulator without saving state. This is the default setting.
Save	Saves the current state to the current Virtual Machine Identifier (VMID) of the emulator. In Visual Studio, this option is disabled if the datastore has set the **protected** flag for this VMID.

See Also

Tasks

How to: Launch the Device Emulator
How to: Launch the Device Emulator Manager
How to: Restore an Emulator OS Image

Other Resources

Device Emulator

Device Emulator
Platform Manager and Emulator Communication
See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Platform Manager, a Windows CE-based technology, manages remote tool connectivity for real or emulated Windows Mobile-based devices in a Platform Builder development environment. When you use a remote tool in a Platform Builder development environment, Platform Manager provides a connection between the development computer and the emulator.

For information on establishing a remote tool connection to the Device Emulator from a Platform Builder development environment, see How to: Configure Remote Tool Communication.

Platform Manager handles only remote tool connectivity, and only for a Platform Builder development environment. All other forms of connectivity, such as downloading an application or run-time image to the Device Emulator or connecting to a development environment for debugging, use the Core Connectivity infrastructure. For more information, see **Core Connectivity Infrastructure** in the Platform Builder Diagnostic Edition.

See Also

Other Resources

Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The following considerations apply to the relationship between the keyboard of the development computer and the keystrokes perceived by the emulator.

Mapping Characteristics

The mapping between the keyboard of the development computer and the Smartphone or Pocket PC emulator keypad is not guaranteed to match. The mapping between desktop keystrokes and clicks on the emulated keypad is provided by the skin files. So, for example, when you click on a * on the emulated Smartphone keypad, the emulator receives an F8 keystroke. Thus, by pressing F8 on your development computer keyboard, you cause the same behavior as if you were clicking on the emulated keypad. On the other hand, if you press SHIFT+8 on the development computer keyboard, a Smartphone emulator OS actually receives SHIFT+8 from what it considers the keypad, and simply ignores the keystroke.

Non-EN-US Development Computer Keyboards

Non-EN-US keyboards do not correctly reproduce characters in the emulators. Changes to character sets are not recognized by the keyboard drivers because the drivers are based simply on the scan codes of the keyboard. To correctly display non-EN-US characters, use internal inputs such as the Pocket PC virtual keyboard or the Smartphone key inputs.

See Also

Other Resources

Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Skins

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

An emulator skin controls the appearance and functionality of the graphical user interface (GUI) for the Device Emulator. Using an emulator skin, you can give the emulator the look and behavior of a specific hardware platform.

You can choose to run the Device Emulator with or without a skin. If you run the emulator with a skin, the skin affects the appearance of the Emulator window. The emulator skin overrides the video display setting for the Device Emulator.

Customizing skins

You can create an emulator skin with an appearance of your choosing and buttons that change in appearance when pressed. For more information, see How to: Customize an Emulator Skin.

Technology

An emulator skin consists of up to three bitmap (BMP) or Portable Network Graphics (PNG) files and a single skin definition (.xml) file that contains Extensible Markup Language (XML). The skin definition file and the associated BMP or PNG files must be in the same directory.

One BMP or PNG file, referred to as a *normal* art file, shows the default appearance of the emulator skin. The normal art file is the only file required by the emulator skin. A second, optional, BMP or PNG file, referred to as a *down* art file, shows the appearance of the emulator skin with all buttons pressed. A third, optional, BMP or PNG file, referred to as a *mapping* file, shows a color code for each button in the emulator skin. The color code for each button is represented in the mapping file as a single-color area that completely covers the button. The user of the emulator skin does not see the colors that you use as codes in the mapping file.

The .xml file describes how the skin works and where to find the BMP or PNG files. The .xml file also defines button actions based on the color code shown in the mapping file. For more information, see Device Emulator XML Schema.

See Also

Reference

Display, Emulator Properties Dialog Box
Device Emulator XML Schema

Other Resources

Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Troubleshooting Connection Issues

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

The following paragraphs are designed to help you resolve situations you might encounter when you try to connect the Device Emulator to the development computer.

Connecting to the Device Emulator

Connecting to Device Emulator Without Network Connection

Connecting to Device Emulator Using ActiveSync

Generating a New MAC Address

Identifying the IP Address of the Emulator

Failure to Open Virtual Switch Driver

Connecting to a Network

See Also

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Configuring and Using the Device Emulator

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

The Device Emulator is designed to be highly configurable. Steps for completing some of the more common tasks appear in the following topics.

In This Section

How to: Change Property Settings

How to: Restore an Emulator OS Image

How to: Generate Saved-State Files

How to: Launch the Device Emulator

How to: Launch the Device Emulator Manager

How to: Download a Run-Time Image to the Emulator

How to: Target a KITL-Enabled Run-Time Image From Visual Studio

How to: Simulate a Warm or Cold Boot

How to: Attach Platform Builder to a Running Emulator

How to: Shut Down the Device Emulator

How to: Configure Remote Tool Communication

How to: Change Ethernet Settings

How to: Map a Directory to the Device Emulator

How to: Remove a Mapped Directory from the Device Emulator

How to: Emulate a Storage Card Using Folder Sharing

How to: Move Files to and from the Device Emulator

How to: Cradle and Uncradle the Device Emulator

How to: Resize RAM in the Device Emulator

How to: Specify a Shared Folder

How to: Specify a Host Key

How to: Specify a Skin File

How to: Change Skin Orientation

How to: Enable Host-Only Networking

How to: Map Serial Ports

How to: Enable Tooltips

How to: Customize an Emulator Skin

How to: Access Smartphone Emulator File System

How to: Display IP Addresses for Smartphone Emulators

Related Sections

Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

How to: Change Property Settings

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

Change Device Emulator property settings using the **Emulator Properties** tabbed dialog box. If an option appears dimmed, it is unavailable in the current environment.

How you display this tabbed dialog box depends on your installation.

To display the dialog box from the Device Emulator File menu

1. Launch the Device Emulator.

   How you launch the Device Emulator depends on your installation. For more information on launching the emulator, see How to: Launch the Device Emulator.

2. On the Device Emulator **File** menu, click **Configure**.

   The **Emulator Properties** tabbed dialog box opens.

To display the dialog box from within the Platform Builder IDE

1. On the Platform Builder **Target** menu, click **Connectivity Options**.
2. In the **Target Device Connectivity Options** dialog box, click **Kernal Service Map**.
3. In the **Download** box, select **Device Emulator**, and then click **Settings**.
4. The **Emulator Properties** tabbed dialog box opens.

To display the dialog box from within the Visual

Studio IDE

1. On the Visual Studio **Tools** menu, click **Options**.
2. In the **Options** dialog box, expand **Device Tools**, and then click **Devices**.
3. Select an emulator from the **Devices** box, and then click **Properties**.
4. In the **<Projectname> Emulator Properties** dialog box, click **Emulator Options**.

   The **Emulator Properties** tabbed dialog box opens.

See Also

Other Resources

Device Emulator User Interface Reference
Configuring and Using the Device Emulator

Device Emulator

How to: Restore an Emulator OS Image

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

To restore an OS image to the emulator, use the Device Emulator Manager.

For more information, see How to: Launch the Device Emulator Manager.

Security Note
Using saved-state files from untrusted sources can pose a security risk.

To restore an emulator OS image

1. On the **File** menu in the **Device Emulator Manager**, click **Restore Image**.
2. In the **Select Image to Restore** dialog box, select the image to be restored, and then click **Open**.

See Also

Tasks

How to: Download a Run-Time Image to the Emulator

Concepts

Saved-State Files

Other Resources

Device Emulator

Device Emulator

How to: Generate Saved-State Files

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use the Imagegen tool to generate saved-state files for the Device Emulator and to perform certain other functions, such as clearing the global saved-state cache.

Note
The Imagegen tool ships with Visual Studio 2005.

To launch Imagegen.exe

1. Open a console window.
2. Navigate to Imagegen.exe, installed by default under \Program Files\Microsoft Visual Studio 8 \SmartDevices\Emulators\DeviceEmulator.
3. Run the file using your choice of options.

   For command-line options, see Imagegen Command Line Reference.

See Also

Concepts

Saved-State Files

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Launch the Device Emulator

See Also

- ☐ Visual Basic (Declaration)

☐ Visual Basic (Usage)
☐ C#
☐ C++
☐ J#
☐ JScript

You can launch the Device Emulator in several different ways, depending on your installation and your personal preferences.

In Visual Studio, clicking **Start** (or **Start Without Debugging**) on the **Tools** menu opens the **Deploy** dialog box, where you can select the emulator, which then launches automatically.

Launching the Device Emulator

To launch the Device Emulator from the command line

1. Open a console window.
2. Navigate to DeviceEmulator.exe, installed by default under \Program Files\Microsoft Device Emulator\1.0.
3. Run the file using your choice of options.

   For command-line options, see Device Emulator Command Line Reference.

To launch the Device Emulator from the Device Emulator Manager

1. Open the Device Emulator Manager.

   This process depends on your installation. Standalone installations of the Device Emulator require you to launch the Device Emulator before you can launch the Device Emulator Manager. For more information, see How to: Launch the Device Emulator Manager.
2. In the Device Emulator Manager window, right-click the emulator you want to launch.
3. On the shortcut menu, click **Connect**.

To launch the Device Emulator from Platform Builder

- See How to: Download a Run-Time Image to the Emulator.

To launch the Device Emulator from the Visual Studio IDE

1. On the Visual Studio **Tools** menu, click **Connect to Device**.
2. In the **Connect to Device** dialog box, select an emulator from the **Devices** box, and then click **Connect**.

   —or—
3. On the Visual Studio **Tools** menu, click **Device Emulator Manager**.
4. In the Device Emulator Manager window, right-click the emulator you want to launch.
5. On the shortcut menu, click **Connect**.

See Also

Other Resources

Device Emulator

Device Emulator

How to: Launch the Device Emulator Manager

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The Device Emulator Manager (DEM) provides a user interface for starting, connecting to, and managing instances of the Device Emulator on the development computer. It provides the same logical functionality as a physical device cradle. For more information, see Device Emulator Manager.

Note
When you click **Exit** on the DEM **File** menu, you disconnect any cradled emulator image. If you simply want to remove the DEM window from the screen, close its window. The DEM continues to run in the background, and is available from the system tray.

Launching With A Datastore Present

If the development computer has a datastore where emulator images are kept, for example, when Visual Studio 2005 is installed, you can open the DEM in any of the following ways.

To launch the Device Emulator Manager from Visual Studio

- On the **Tools** menu, click **Device Emulator Manager**.

To launch the Device Emulator Manager from Windows Explorer

1. Navigate to Program Files\Microsoft Device Emulator\1.0.
2. Open dvcemumanager.exe.

To launch the Device Emulator Manager from the command line

1. Navigate to Program Files\Microsoft Device Emulator\1.0.
2. Type **dvcemumanager**.

Launching When No Datastore Is Present

If the development computer contains no datastore, for example, in a standalone installation of the Device

Emulator and DEM, you must launch an instance of the Device Emulator before you can open the DEM.

To launch the Device Emulator Manager when no datastore is present

1. Launch an instance of the Device Emulator.

   For more information, see How to: Launch the Device Emulator.

2. Launch the Device Emulator Manager in any of the ways listed earlier for launching with a datastore present.

3. On the Device Emulator Manager **Actions** menu, click **Refresh**.

See Also

Other Resources

Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

You can download a run-time OS image to the Device Emulator using Platform Builder, Visual Studio, or the command line. For more information on launching the Device Emulator from the command line, see How to: Launch the Device Emulator.

Security Note
Using saved-state files from untrusted sources can pose a security risk.

To download a run-time image to the Device Emulator using Visual Studio

1. On the Visual Studio **Tools** menu, click **Connect to Device**.

2. In the **Connect to Device** dialog box, select the OS image you want, and then click **Connect**.

To download a run-time image to the Device Emulator using Platform Builder

1. On the Platform Builder **Target** menu, click **Connectivity Options**.

2. In the left pane, select **Kernel Service Map**.
3. In the **Download** box, click **Device Emulator**.
4. Click **Settings**.
5. In the **Emulator Properties** dialog box, modify settings for the Device Emulator, and then click **OK**.
6. In the **Transport** box, select **Ethernet**.
7. Click **Core Service Settings**. In **KITL Settings**, select any of the following settings you want:
    - **Enable KITL**
    - **Enable access to desktop files**
    - **Clear memory on soft reset**
8. Click **Apply**.
9. On the **Target** menu, click **Attach Device**.

    This step assumes you have an image open in your workspace.

See Also

Reference

Display, Emulator Properties Dialog Box
General, Emulator Properties Dialog Box
Network, Emulator Properties Dialog Box
Peripherals, Emulator Properties Dialog Box

Other Resources

Configuring and Using the Device Emulator

Device Emulator
How to: Target a KITL-Enabled Run-Time Image From Visual Studio
See Also

If you create a run-time image in Windows Mobile and download the image to the Device Emulator, or to a physical device, you can connect to that image from Visual Studio 2005 by associating the device image with an installed SDK. The following steps guide you through configuring the target device and required services.

Note
For an Ethernet connection only, you cannot connect to a target device unless the development computer and target device are on the same subnet.

To perform the following procedures:

- Visual Studio 2005, Platform Builder 5.0 or later, and a KITL-enabled SDK must all be installed on the development computer.
- You must have compiled a run-time image with KITL enabled, and have opened the run-time image from Platform Builder.

To configure the device

1. On the Platform Builder **Target** menu, click **Connectivity Options**.
2. Click **Add Device**.
3. In the **Associated OS Design/SDK (Optional)** drop-down list, select the name of the SDK platform you want to associate with the target device.
4. In the **New target device name** box, type a name for the target device.

   -or-

   If you want to use an existing device, verify that the **New target device name** box is blank.
5. Click **Add**.

To configure kernel services

1. Select **Kernel Service Map**.
2. In the **Target Device** box, select the name of the target device you want to connect.
3. In the **Download** box, select a download service.
4. To the right of the **Download** box, click **Settings**.
5. Configure the settings for the download service, and then click **OK**.
6. In the **Transport** box, select a kernel transport.
7. To the right of the **Transport** box, click **Settings**.
8. Configure the settings for the transport service, and then click **OK**.
9. Select one of the following settings in the **Debugger** box:
   - If your run-time image includes support for KdStub, the kernel debugger stub, click **KdStub**.
   - If your run-time image includes support for hardware-assisted debugging, select the appropriate hardware-assisted debugger.
   - If your run-time image does not include support for a debugger, click **None**.

To configure core services

1. Click **Core Service Settings**.
2. Select one of the following settings under **Download Image**:
   - To instruct Platform Builder to download a run-time image each time Platform Builder connects with the target device, select **Always**.
   - To instruct Platform Builder to download a run-time image if the run-time image on the development computer differs from the run-time image on the target device, select **Only if image changed**.
   - To instruct Platform Builder to download a run-time image only the first time Platform Builder connects with the target device, select **Only on initial download**.
   - To instruct Platform Builder to jump to the run-time image on the target device, select **Never (jump to image only)**.
3. Click **Apply**.

See Also

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Simulate a Warm or Cold Boot

See Also

You can use the emulator to simulate either a warm boot or a cold boot of your mobile device.

To simulate a warm boot

- On the Device Emulator **File** menu, click **Reset**, and then click **Soft**.

  A soft restSoft Reset restarts the device while leaving the current file store and registry information intact.

To simulate a warm boot using Platform Builder

- On the **Target** menu item in Platform Builder, click **Reset Device**.

To simulate a cold boot

- On the Device Emulator **File** menu, click **Reset**, and then click **Hard**.

  A hard reset reloads the factory defaults, including the OS image, discarding any user changes.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

To attach Platform Builder to an instance of the Device Emulator that is running, you must boot the Device Emulator using the eboot.nb0 image. If the Device Emulator is started using a kernel .bin file, the run-time image boots without the kernal independent transport layer (KITL) and cannot be attached to Platform Builder for a virtual machine session.

To attach to a Device Emulator that is running

1. Launch the Device Emulator from the command line using the eboot.nb0 image file.

   For more information, see How to: Launch the Device Emulator.

2. After the Device Emulator opens, open Platform Builder.
3. On the **Target** menu, click **Connectivity Options**.
4. In the left pane, select **Kernel Service Map**.
5. In the **Download** box:
   - Select **None** if the run-time image is booted; or
   - Select **Ethernet** to use Ethernet to download the run-time image.
6. Click **Settings**.

7. In the **Emulator Properties** dialog box, modify settings for the Device Emulator, and then click **OK**.
8. In the **Transport** box, select **Ethernet**.
9. Click **Core Service Settings**. In **KITL Settings**, select any of the following settings that you want:
    - Enable KITL.
    - Enable access to desktop files.
    - Clear memory on soft reset
10. Click **Apply**.
11. On the **Target** menu, click **Attach Device**.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

You can use either the Device Emulator Manager or the Device Emulator itself to shut down the emulator.

- If you shut down from the emulator window, the emulator prompts for whether or not you want to save the state of the emulator. For more information on saving state, see Saved-State Files.
- If you shut down the emulator from the Device Emulator Manager, the state is not saved.

To shut down the Device Emulator from the emulator window

- In the Device Emulator window, click **File**, and then click **Exit**.

  -or-

- In the Device Emulator window, click Close (**X**) in the upper right corner.

To shut down the Device Emulator from the Device Emulator Manager

1. In the Device Emulator Manager window, right-click the instance of the emulator to be shut down.
2. On the shortcut menu, click **Shutdown**.

See Also

Concepts

Device Emulator Manager

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Configure Remote Tool Communication

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

You can use remote tools, such as Remote Registry Editor or Remote Spy, to test a run-time image on the Device Emulator. How you launch remote tools varies with your installation.

Visual Studio 2005

To launch remote tools

- Click **Start** on the Windows desktop, point to **All Programs**, point to **Microsoft Visual Studio 2005**, click **Visual Studio Remote Tools**, and then select one of the remote tools from the menu.

Platform Builder

To configure a remote tool for Device Emulator connectivity

1. Download the run-time image you want to use. For more information, see How to: Download a Run-Time Image to the Emulator.
2. On the Platform Builder **Tools** menu, select any remote tool.

3. On the **Connection** menu in the **Remote Tool** window, click **Configure Windows CE Platform Manager**.
4. Expand the **Windows CE Default Platform** node.

   —or—

   Expand the node for the devices you want to configure.
5. Select **Default Device**, and then click **Properties**.

   —or—

   If you created a device, select the entry for your device, and then click **Properties**.
6. In the **Transport** box, select a transport, and then click **Configure**.
7. Customize the settings for the transport, and then click **OK**.
8. In the **Startup Server** box, select the startup server, and then click **Configure**.
9. Customize the settings for the startup server, and then click **OK**.
10. Click **OK**.

See Also

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Change Ethernet Settings

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

You can change the network card emulation that provides Ethernet support for the Device Emulator from the Device Emulator window, from Platform Builder, or from Visual Studio.

Note
Ethernet support is available only if you install the Virtual Switch Driver. You can download this driver from the Mobile Developer Center.

To change Ethernet support settings using the Device Emulator window

1. On the Device Emulator **File** menu, click **Configure**.
2. In the **Emulator Properties** dialog box, click the **Network** tab.
3. Select the network card you want to emulate, and then click **OK**.

To change Ethernet support settings using Platform Builder

1. On the Platform Builder **Target** menu, click **Connectivity Options**.
2. Click **Kernel Service Map**.
3. In the **Download** box, verify that **Device Emulator** is selected, and then click **Settings**.
4. In the **Emulator Properties** dialog box, select the **Network** tab.
5. Select the network card you want to emulate, and then click **OK**.
6. Choose **Apply**.

To change Ethernet support settings using Visual Studio

1. On the Visual Studio **Tools** menu, click **Options**.
2. In the **Options** dialog box, expand **Device Tools**, and then click **Devices**.
3. Select the emulator whose settings you want to change, and then click **Properties**.
4. In the **Emulator Properties** dialog box, click the **Network** tab.
5. Select the network card you want to emulate, and then click **OK**.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

Device Emulator
How to: Map a Directory to the Device Emulator
See Also

☐ Visual Basic (Declaration)

☐ Visual Basic (Usage)
☐ C#
☐ C++
☐ J#
☐ JScript

You can connect the Device Emulator to a directory on your development computer as if the directory were a drive mounted on the Device Emulator. In the file system for the Device Emulator, the connected directory appears at the root level as Storage Card.

You can connect the Device Emulator to only one directory at a time.

Security Note
Any use of the folder-sharing feature poses a potential security risk. For example, do not enable the sharing of directories that contain confidential or sensitive information.

To map a directory to the Device Emulator

1. On the Device Emulator, run a Windows CE OS that supports the ability to connect to a directory on the development computer.
2. On the Device Emulator **File** menu, click **Configure**.
3. In the **Emulator Properties** dialog box, select the **General** tab.
4. In the **Shared folder** box, type the path and folder name for the directory you want to map.

   —or—

   Click the **Browse** button, navigate to the directory that you want to share, and then click **OK**.
5. Click **OK**.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

Device Emulator
How to: Remove a Mapped Directory from the Device Emulator
See Also

☐ Visual Basic (Declaration)
☐ Visual Basic (Usage)
☐ C#
☐ C++

☐ J#
☐ JScript

If you previously mapped a directory to the Device Emulator, you can remove the mapping by using the **Emulator Properties** dialog box. This dialog box is available using Platform Builder and Visual Studio, as well as using the Device Emulator window as described in the following task. For more information, see How to: Change Property Settings.

To remove a mapped directory from the Device Emulator

1. On the Device Emulator **File** menu, click **Configure**.
2. In the **Emulator Properties** dialog box, select the **General** tab.
3. In the **Shared folder** box, delete the path and folder.

See Also

Tasks

How to: Map a Directory to the Device Emulator

Other Resources

Configuring and Using the Device Emulator

Device Emulator
How to: Emulate a Storage Card Using Folder Sharing
See Also

☐ Visual Basic (Declaration)
☐ Visual Basic (Usage)
☐ C#
☐ C++
☐ J#
☐ JScript

When a folder is shared to the emulator, it appears as a storage card. The emulator responds to the shared folder storage card as if it were a real device. You can also stop folder sharing when you no longer want to work with an emulated storage card.

Security Note
Any use of the folder-sharing feature poses a potential security risk. For example, do not enable sharing of directories that contain confidential or sensitive information.

To activate folder sharing

1. On the Device Emulator **File** menu, click **Configure**.

2. Select the **General** tab.
3. In the **Shared folder** box, type the path and folder name for the directory you want to use as an emulated storage card.

   -or-

   Click the **Browse** button, navigate to the directory that you want to use as an emulated storage card, and then click **OK**.
4. Click **OK**.

To stop folder sharing

1. On the Device Emulator **File** menu, click **Configure**.
2. Select the **General** tab.
3. In the **Shared folder** box, delete the path and folder name for the directory you used as an emulated storage card.
4. Click **OK**.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

To move files to and from the Device Emulator, use a shared folder.

For information about creating a shared folder, see How to: Map a Directory to the Device Emulator.

Security Note
Any use of the folder-sharing feature poses a potential security risk. For example, do not enable the sharing of directories that contain confidential or sensitive information.

To use a shared folder to move files to or from the emulator

1. On the development computer, create a shared folder.
2. To move a file to or from the Device Emulator, move or copy the file into or out of the shared folder directory.

See Also

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Cradle and Uncradle the Device Emulator

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use the Device Emulator Manager to cradle and uncradle the emulator, simulating the cradling and uncradling of a physical device.

To perform these tasks, an instance of the Device Emulator must be connected.

To cradle and uncradle the Device Emulator

1. Launch the Device Emulator Manager.

   For more information on launching the Device Emulator Manager, see How to: Launch the Device Emulator Manager.

2. In the **Device Emulator Manager** window, right-click the connected instance of the emulator.
3. On the shortcut menu, click **Cradle** or **Uncradle**.

See Also

Tasks

How to: Launch the Device Emulator Manager

Concepts

Device Emulator Manager

Other Resources

Configuring and Using the Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

How to: Resize RAM in the Device Emulator

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

You can change the size of RAM to any value supported by the OS image. Use this feature to more closely mimic RAM conditions in the physical device you are emulating.

If you change RAM size while the emulator is running, you must restart the emulator for the changes to take effect.

Some images might not support the resizing of RAM.

To resize RAM using Visual Studio

1. On the Visual Studio **Tools** menu, click **Options**, click **Device Tools**, click **Devices**, select the emulator whose RAM you want to change, click **Properties**, and then click **Emulator Options**.
2. On the **General** tab, select **Specify RAM size**.

To resize RAM using Platform Builder

1. On the Platform Builder **Target** menu, click **Connectivity Options**.
2. Select **Kernel Services Map**.
3. In the **Download** box, select **Device Emulator**.
4. To the right of the **Download** box, click **Settings**.
5. On the **General** tab, select **Specify RAM size**.

To resize RAM using the command line

- Launch the Device Emulator from the command line using the **/memsize** switch.

  For more information, see How to: Launch the Device Emulator.

See Also

Reference

Device Emulator Command Line Reference

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Specify a Shared Folder

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use shared folders to emulate storage cards, move files, and so forth.

Security Note
Any use of the folder-sharing feature poses a potential security risk. For example, do not enable the sharing of directories that contain confidential or sensitive information.

To specify a shared folder

1. On the Device Emulator **File** menu, click **Configure**.
2. On the **General** tab, specify the shared folder either by inserting the name or by navigating to the folder using the **Ellipsis** (**…**) button.

See Also

Tasks

How to: Launch the Device Emulator
How to: Move Files to and from the Device Emulator
How to: Emulate a Storage Card Using Folder Sharing
How to: Remove a Mapped Directory from the Device Emulator

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Specify a Host Key

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

While an emulator is running, you can use the keyboard of your development computer to control certain emulator actions, such as displaying its shortcut menu. You implement this technique by predefining key combinations that include a host key and one or more other keys. The emulator recognizes these key combinations as commands. For more information, see Device Emulator Key Combinations.

To specify a host key

1. On the Device Emulator **File** menu, click **Configure**.
2. On the **General** tab, select the host key.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

How to: Specify a Skin File

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

You specify a skin for the Device Emulator by specifying an XML skin definition file.

For more information on skin definition files, see Device Emulator XML Schema.

To specify a skin file

1. On the Device Emulator **File** menu, click **Configure**.

2. On the **Display** tab, select **Skin**.
3. Insert the skin file name, or navigate to the file by clicking the **Ellipsis** (**…**) button.

See Also

Tasks

How to: Launch the Device Emulator
How to: Customize an Emulator Skin

Reference

Display, Emulator Properties Dialog Box

Concepts

Skins

Other Resources

Configuring and Using the Device Emulator

Device Emulator

How to: Change Skin Orientation

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use this feature to change from portrait to landscape, or any degree of rotation available in the **Emulator Properties** dialog box.

To change skin orientation

1. On the Device Emulator **File** menu, click **Configure**.
2. On the **Display** tab, select the number of degrees of rotation.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

How to: Enable Host-Only Networking

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Host-only networking provides a network connection between the desktop computer and the Device Emulator. This network is completely contained within the host.

To enable host-only networking

1. On the Device Emulator **File** menu, click **Configure**.
2. On the **Network** tab, select **Host-only networking**.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

How to: Map Serial Ports

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The Device Emulator provides three serial ports, virtual COM ports, that you can map to COM ports on the development computer.

To map serial ports

1. On the Device Emulator **File** menu, click **Configure**.
2. On the **Peripherals** tab, specify the mapping.

   For more information, see Peripherals, Emulator Properties Dialog Box.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Tooltips on the emulator display labels for skin features, such as Soft Key 1.

To enable tooltips

1. On the Device Emulator **File** menu, click **Configure**.
2. On the **Display** tab, click **Enable tooltips**.

See Also

Tasks

How to: Launch the Device Emulator

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The following steps describe how to develop a custom skin for the Device Emulator, and then how to apply the skin to an emulator in Platform Builder and Visual Studio.

To develop a custom skin

1. Create a bitmap (BMP) or Portable Network Graphics (PNG) file that shows the default appearance of the emulator skin.
2. Create a BMP or PNG file that shows the appearance of the emulator skin with all buttons pressed.
3. Create a BMP or PNG file that shows the area of each button filled with a single unique color.
4. If you want the appearance of each button to change independently of the other buttons in the emulator skin, use a different color to fill the area for each button.

   Note
   For best visibility, do not to use white or black to fill this area.
5. Create an .xml skin-definition file.

   For more information, see Device Emulator XML Schema.
6. Save the three BMP or PNG files and the .xml file to a single directory.

To use the skin in a Platform Builder project

1. On the Platform Builder **Target** menu, click **Connectivity Options**.
2. In the **Download** box, select **Device Emulator**.
3. To the right of the **Download** box, click **Settings**.
4. In the **Emulator Properties** dialog box, click the **Display** tab.
5. Select **Skin**, and then navigate to the location of your custom skin-definition .xml file.

To use the skin in a Visual Studio project

1. On the Visual Studio **Tools** menu, click **Options**.
2. Expand the **Device Tools** node, and then click **Devices**.
3. In the **Devices** box, select the emulator you want to add the skin to, and then click **Properties**.
4. In the **<DeviceName> Emulator Properties** dialog box, click **Emulator Options**.

5. In the **Emulator Properties** dialog box, click the **Display** tab.
6. Select **Skin**, and then navigate to the location of your custom skin-definition .xml file.

See Also

Reference

Device Emulator XML Schema

Concepts

Skins

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The Smartphone emulator has no File Explorer. Use the following technique to access the Smartphone file system.

Note
This technique is available only from Visual Studio.

To access the Smartphone emulator file system

1. On the Visual Studio **Tools** menu, click **Device Emulator Manager**.
2. In the **Available Emulators** pane, select the emulator whose file system you want to access.
3. On the Device Emulator Manager **Actions** menu, click **Connect**.

   An icon is displayed beside the selected emulator indicating that a connection has been made.
4. Right-click the selected emulator, and then click **Cradle**.

   The icon changes to show the emulator is cradled.
5. Open ActiveSync.
6. On the ActiveSync **File** menu, click **Connection Settings**.

7. Select **Allow connections to one of the following**.
8. Select **DMA** from the list of ports, and then click **OK**.

   ActiveSync now initiates a partnership with the emulator. Follow the instructions provided by the **New Partnership** wizard.

   Note
   If a partnership is not initiated automatically, click **Connect** in the **Connection Settings** dialog box, a then follow the prompts in the **Get Connected** wizard.
9. After you have completed the ActiveSync partnership steps, click **Explore** on the ActiveSync toolbar to access the emulator file system.

   Note
   If you close the Device Emulator Manager or close the emulator, the ActiveSync connection also clos

See Also

Other Resources

Configuring and Using the Device Emulator

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Discovering IP addresses for Smartphone emulators requires a programmatic approach. The following steps provide for creating and running one such routine.

To create the routine

1. In Visual Studio, open a new C# Smartphone empty project.
2. In **Solution Explorer**, right-click the project, point to **Add**, and then click **New Item**.
3. Select **Code File**, and then click **Add**.

   The code editor opens with a blank page.
4. Copy the following code block onto the editor page.

Copy Code

```
using System;
using System.Net;
using System.Text;
```

```
using System.Windows.Forms;

public class GetAddress
{
    /// <summary>
    /// A sample application that displays a list of IP addresses
    /// that are bound to the current device.
    /// </summary>

    static void Main()
    {
        try
        {
            IPHostEntry IPHost = Dns.Resolve(Dns.GetHostName());
            IPAddress[] addressList = IPHost.AddressList;

            if (addressList.Length > 0)
            {
                StringBuilder address = new StringBuilder();
                foreach (IPAddress a in addressList)
                {
                    address.Append(a.ToString());
                    address.Append(" ");
                }
                MessageBox.Show(address.ToString(), "IP Addresses");
            }

            else
                MessageBox.Show("Unable to determine network address", "Error");
        }

        catch (Exception)
        {
            MessageBox.Show("Unable to determine network address", "Error");
        }
    }
}
```

5. In **Solution Explorer**, right-click **References**, and then click **Add Reference**.
6. Select **System.Windows.Forms**, and then click **OK**.
7.

To run the routine

1. On the **Debug** menu, click **Start Debugging**.
2. In the **Deploy** dialog box, select the Smartphone emulator whose IP addresses you want to display.
3. Click **Deploy**.

   The application displays the IP addresses.

See Also

Tasks

Troubleshooting Connection Issues

Other Resources

Configuring and Using the Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Device Emulator Reference

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

These topics provide reference information for the Device Emulator and skins.

In This Section

Device Emulator Command Line Reference

Provides syntax for launching the Device Emulator from the command line.

Imagegen Command Line Reference

Provides syntax for launching the Imagegen tool.

Device Emulator XML Schema

Provides an XML model for developing custom skin files for the Device Emulator.

Device Emulator Key Combinations

Specifies key combinations on the development computer that you can use to control some aspects of a running emulator.

Device Emulator User Interface Reference

Describes the dialog boxes used to configure the Device Emulator.

Related Sections

Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Device Emulator Command Line Reference

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#

- ☐ C++
- ☐ J#
- ☐ JScript

Launch the Device Emulator from the command line using the following syntax. The os_image_file_name is required unless you use the /s switch.

Note
Not all SDKs support all options. Check your SDK documentation for details.

```
DeviceEmulator os_image_file_name [/a] [/c] [/f [featurevalue]]
   [/flash [flash_file_name]] [/h] [/hostkey keyname]
   [/language LangID] [/memsize size] [/n [macaddress]]
   [/nosecurityprompts] [/p [macaddress]] [/r ROM_address]
   [/rotate angle] [/s save-state_file_name]
   [/sharedfolder directoryname] [/skin skin_file_name] [/tooltips
        state] [/u0 serialport] [/u1 serialport] [/u2 serialport]
      [/video <width>x<height>x<bit depth>] [/vmid {GUID}]
   [/vmname name] [/z]
```

Parameters

Parameter	Description
os_image_file_name	Specifies the path and file name (*.bin or *.nb0) of the kernel image the emulator is to use. Required, unless you launch using the /s switch.

Switches

See Also

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Imagegen Command Line Reference

See Also Example

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use Imagegen.exe to generate and store Device Emulator images.

Note
You need administrator permission to use any option that affects the global cache. The global cache is located under \Documents and Settings\All Users\Application Data\Microsoft\Device Emulator.

Imagegen.exe ships with Visual Studio 2005, and is located by default at \Program Files\Microsoft Visual Studio 8\SmartDevices\Emulators\DeviceEmulator.

```
Imagegen [/i] [/g] [/l] [/u]
```

Parameters

Option	Description
/i	The GUID that identifies the device whose state is to be saved.
/g	Targets the global saved-state cache instead of the per-user cache.
/l	Sets the locale ID for the datastore. If you do not specify an ID, the ID value defaults to 1033.
/u	Clears the global cache.

Example

See Also

Device Emulator
Device Emulator XML Schema
See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use the following reference as a model for developing custom skin files for the Device Emulator.

Note
You can use PNG as well as BMP image files.

Code

Remarks

See Also

Device Emulator
Device Emulator Key Combinations
See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use the keyboard of your development computer to control certain actions on the Device Emulator while it is running.

For more information on viewing or changing the Host Key, see How to: Specify a Host Key.

Action	Key combination
Shutdown	Host Key+F4
Display **File** menu	Host Key+Alt+F
Display **Help** menu	Host Key+Alt+H
Display shortcut menu	Host Key+Alt+Spacebar

See Also

Other Resources

Device Emulator Reference

Device Emulator
Device Emulator User Interface Reference
See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

This section describes user interface elements for setting Device Emulator options.

Note
The Device Emulator Manager dialog box is available only in the Visual Studio environment.

In This Section

Display, Emulator Properties Dialog Box

Provides for altering display characteristics such as height, width, and orientation.

General, Emulator Properties Dialog Box

Provides for specifying basic options such as OS image, ROM address for start of execution, and flash memory files.

Network, Emulator Properties Dialog Box

Provides for binding virtual network adapters to real network cards on the host.

Peripherals, Emulator Properties Dialog Box

Provides for changing settings for peripheral devices connected through serial ports.

Device Emulator Manager Dialog Box

Lists available emulators and provides for connecting to them.

See Also

Other Resources

Device Emulator

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

Display, Emulator Properties Dialog Box

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use this tab to specify the appearance characteristics of the emulator.

How you display this tabbed dialog box depends on your installation. For more information, see How to: Change Property Settings.

You can specify screen sizes and color depth only prior to cold-booting the emulator. That is, you cannot change these properties at run time.

Using different screen sizes can lead to emulator malfunction. The following screen sizes are recommended:

Device	Screen size
PocketPC	240x320
PocketPC VGA	480x640
Smartphone	176x220
SmartPhone QVGA	240x320

Skin

If you select **Skin**, click the ellipsis button (**…**) to locate the skin file. If you do not want the skin to be displayed, select **Video**. If you select **Video** you must complete all video settings.

For more information on skins, see Emulator Skin in the Windows CE documentation.

Screen width

Minimum width is 64 pixels, maximum is 800. The number must be an even number.

Screen height

Minimum width is 64 pixels, maximum is 800. The number must be an even number.

Color depth

Select from the drop-down list in multiples of 8.

Orientation

Specifies orientation of 0, 90, 180, or 270 degrees. You can change the value at run time. The emulator automatically changes orientation when the guest OS is switched from Portrait to Landscape mode.

For more information, see your SDK documentation.

Zoom 2x

Stretches the skin on both axes, resulting in an image four times as large as the original. You can change this setting at run time.

Always on top

Maintains the emulator on top of the z order in the development environment. You can change this setting at run time.

Enable tooltips

Hovering over any menu or button displays a ToolTip. This feature is available only if a skin file is used and the ToolTip is defined in it.

See Also

Tasks

How to: Change Property Settings

Reference

General, Emulator Properties Dialog Box
Network, Emulator Properties Dialog Box
Peripherals, Emulator Properties Dialog Box

Other Resources

Device Emulator User Interface Reference

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.

Device Emulator

General, Emulator Properties Dialog Box

See Also

- Visual Basic (Declaration)
- Visual Basic (Usage)
- C#
- C++
- J#
- JScript

Use this tab to set properties of the emulator itself, such as which kernel image to load.

How you display this tabbed dialog box depends on your installation. For more information, see How to: Change Property Settings.

OS Image Name

Specifies the name of the kernel image.

Clicking the ellipsis button opens a file dialog where you can select a .bin (OS ROM image) file.

If the emulator has already booted, this field is read-only. To change the image file, you need to restart the emulator.

Security Note
Using saved-state files from untrusted sources can pose a security risk.

Specify ROM Image Address

Specifies a hex address inside the ROM image at which execution will start. This feature is useful when you are working with custom images.

If you want to change the address, you must restart the emulator for the change to take effect.

Specify RAM size

Sets size of memory in megabytes. Default value is 64 megabytes.

If you want to change RAM size, you must restart the emulator for the change to take effect.

Flash Memory File

(Read-only at run time.) Specifies a file containing a saved image of NAND flash. If the file does not exist, the flash is initialized with a blank formatted image. You can save the state of NAND flash by using the **Flash/Save** menu on the emulator.

You can use the same flash file with more than one OS image.

You can initialize the NAND flash state only from a cold boot. Thus, the saved state files internally contain the current state of NAND flash.

Note
Not all SDKs support NAND flash. When NAND Flash is not supported, the **Save** menu item on the l menu is dimmed and not available.

Host key

Specifies a key used to send chords, for example, ALT+TAB, to the host operating system.

Shared folder

Specifies a directory to be mounted as a shared folder.

See Also

Reference

Display, Emulator Properties Dialog Box
Network, Emulator Properties Dialog Box
Peripherals, Emulator Properties Dialog Box

Other Resources

Device Emulator User Interface Reference

Device Emulator
Network, Emulator Properties Dialog Box
See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use this tab to bind virtual network adapters to real network cards on the host system.

How you display this tabbed dialog box depends on your installation. For more information, see How to: Change Property Settings.

If your development computer has no network connection, install the Microsoft Loopback Adapter, select the **Loopback Adapter** network card, and then select **Host-only Networking**. For more information, see "Installing the Microsoft Loopback Adapter" in the MSDN Online Library.

Enable NE2000 PCMCIA Network Adapter and bind to

Specifies what network card this adapter will bind to. The default card is **Connected network card**. Either the OS or the Kernel Independet Transport Layer (KITL) can use the NE2000 for general networking.

Use this default setting unless you have more than one network card or if you are using the Microsoft Loopback Adapter.

Enable CS8900 Network Adapter and bind to

Enable this adapter only if you are using a KITL-enabled image. Windows Mobile™ OS images cannot use the CS8900 card for general networking because they do not include CS8900 drivers.

Host-only Networking

Creates a network totally contained within the host. For typical work, leave this check box cleared to maintain connectivity with Visual Studio.

See Also

Reference

General, Emulator Properties Dialog Box
Display, Emulator Properties Dialog Box
Peripherals, Emulator Properties Dialog Box

Other Resources

Device Emulator User Interface Reference

Device Emulator

Peripherals, Emulator Properties Dialog Box

See Also

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

Use this tab to change settings for peripheral devices connected through serial ports.

How you display this tabbed dialog box depends on your installation. For more information, see How to: Change Property Settings.

Serial port options for each of the three ports are as follows:

- Do not map.
- All available and enabled COM ports on the host system.

If a serial port setting contains a COM port, then the COM port must be unique relative to the other serial port settings. If you select a COM port to which nothing is connected, the emulator might not boot.

Valid mappings are all available and enabled COM ports on the host system.

Serial port 0

The emulator COM1 port, used by the OS for debug output.

Serial port 1

The emulator COM2 port, used as the serial port for the product.

Enabling or disabling **KITL** does not change the behavior of the port.

Serial port 2

The emulator COM3 port, used as the IRDA port.

Create text console window for Serial port 0

Opens a console window to show all input and output of the port selected in Serial port 0. The output of **OutputDebugString** goes to this console.

See Also

Tasks

How to: Map Serial Ports

Reference

General, Emulator Properties Dialog Box
Display, Emulator Properties Dialog Box
Network, Emulator Properties Dialog Box

Other Resources

Device Emulator User Interface Reference

- ☐ Visual Basic (Declaration)
- ☐ Visual Basic (Usage)
- ☐ C#
- ☐ C++
- ☐ J#
- ☐ JScript

The Device Emulator Manager is available from the Visual Studio 2005 **Tools** menu.

Available Emulators

Lists the emulators available on the development computer.

You can connect to an emulator by right-clicking it in the tree view, and then clicking **Connect** on the shortcut menu.

Refresh

Refreshes the tree view.

See Also

Other Resources

Device Emulator User Interface Reference

To make a suggestion or report a bug about Help or another feature of this product, go to the feedback site.