



## **MTP Responder Deployment Guide**

Writers: Mark McLemore

Technical Reviewer: Alex Bussmann, Andy Haslam

Published: January 2012

Applies To: Windows Embedded Compact 7

### **Abstract**

This paper is a step-by-step guide to implementing MTP Responder on your device. MTP Responder connects to Device Stage on a Windows 7 computer for device discovery and file synchronization or transfer. This paper describes:

- Development prerequisites and tasks
- Catalog items and registry settings
- How to provide OAL support for Device Stage
- How to customize your Device Stage presentation

For information about MTP Responder internal architecture, optimization and source code modification, see [Inside MTP Responder](http://go.microsoft.com/fwlink/?LinkID=210328) (<http://go.microsoft.com/fwlink/?LinkID=210328>).

# Contents

Introduction .....	4
Device Stage Support .....	4
Windows Logo Certification .....	5
MTP Responder Components .....	6
MTP Responder Stack.....	6
MTP Storage .....	6
MTP Transports.....	6
Prerequisites.....	7
Windows Embedded Compact 7 .....	7
Metadata Tools.....	7
Windows 7 .....	7
Development Tasks .....	7
Device Stage Implementation.....	8
Step 1 Verify System Requirements .....	8
Windows 7 and Windows Vista .....	8
Hardware Requirements.....	8
Software Requirements .....	9
Windows XP .....	9
Hardware Requirements.....	9
Software Requirements .....	9
Step 2 Add MTP Responder Catalog Items to Your OS Design .....	9
Step 3 Configure Registry Settings in Your OS Design .....	10
Device Information.....	10
Transport Information.....	13
Device Metadata Service .....	14
MTP Storage Settings.....	15
Storage Location .....	15
Default Storage .....	16
Media MDB Settings .....	17
Thumbnail Storage .....	17
Step 4 Provide OAL Support for Device Stage.....	19
Manufacturer Name .....	20
Device ID.....	20
Battery Level.....	20
Free Storage Space.....	21
File Time Stamps.....	21
Step 5 Optionally Handle Session Notifications .....	21

Step 6 Optionally Handle MTP IP Authentication .....	23
Step 7 Optionally Handle Power Management Scenarios .....	23
Step 8 Create a Presentation for Windows 7 .....	24
Device Icon.....	24
Baseline Presentation .....	24
Remove the Media Synchronization Task .....	25
Custom Presentation .....	26
Step 9 Test Your Presentation.....	27
Baseline Presentation .....	27
Custom Presentation .....	27
Step 10 Complete Windows Logo Testing .....	28
Step 11 Submit Your Presentation to Windows Quality Online Services.....	28
Optimizations and Modifications .....	28
Conclusion .....	29
Additional Resources .....	29

# Introduction

---

MTP Responder for Windows Embedded Compact 7 provides enhanced Media Transfer Protocol (MTP) functionality to support Microsoft Device Stage technology. Device Stage, introduced in Windows 7, offers a central location in Windows for your users to discover and use their devices. When you include Device Stage support in your device, users can get detailed information about the device, copy files to and from the device, run device-specific tasks, read a product manual, and buy accessories for the device, all in one convenient location. Device Stage provides these capabilities without requiring the user to download or install software. For more information about Device Stage, see the [Windows Device Experience website](http://go.microsoft.com/fwlink/?LinkId=183519) (<http://go.microsoft.com/fwlink/?LinkId=183519>).

Device Stage communicates with your device by using MTP; therefore, a device must implement MTP to support Device Stage. Using MTP, an application on a computer—the MTP initiator, controls a portable device—the MTP responder, and transfers digital media content and metadata between the computer and the device. The MTP initiator and the MTP responder communicate by using a communications link, or MTP transport, typically a USB or TCP/IP connection between the computer and the device. Devices participate in Device Stage by implementing an MTP responder that can handle Device Stage requests from an MTP initiator over one or more MTP transports. The MTP Responder functionality provided in Windows Embedded Compact 7 is specifically designed to support the MTP initiator that is included in Device Stage.

This guide describes the Device Stage support provided by MTP Responder, the functionality of each MTP Responder catalog item, MTP Responder limitations, and prerequisites for including MTP Responder in your device. A detailed step-by-step Device Stage implementation section explains how to install MTP Responder, how to configure device registry settings, how to add OEM adaptation layer (OAL) support, how to create presentation elements, and how to test your device with Device Stage.

MTP Responder for Windows Embedded Compact 7 is based on the MTP responder in the Windows 7 Portable Device Enabling Kit (DEK) for MTP, version 7R2. This kit contains the source code for the reference implementations of an MTP initiator and an MTP responder. We derived MTP Responder from the reference MTP responder in this kit, and added functionality so that Windows Embedded Compact 7 devices support Device Stage operations on Windows 7 host computers. Windows Embedded Compact 7 devices also support Windows Media Player media synchronization and Windows Explorer operations on Windows 7, Windows Vista, and Windows XP hosts.

## Device Stage Support

When you include MTP Responder for Windows Embedded Compact 7 in your device, you can choose from one of two supported levels of Device Stage presentation: **baseline** or **custom**. The baseline presentation is the most basic device presentation level; the customized presentation offers a richer set of features for using your device.

The baseline presentation displays to the user a device icon, some customized information about your device, and basic status information. MTP Responder gives your device the functionality to communicate this essential information about the device and its contents to Device Stage. Using this information, MTP Responder provides the following fundamental Device Stage capabilities:

- **View battery status** Device Stage displays battery status information, if your device reports this information.
- **View storage space** Device Stage displays the amount of storage space available on your device, if your device reports this information.
- **Browse files** File browsing capability lets a user manage files and folders on a device using an Explorer window on the computer, and transfer files between the device and the computer.
- **Manage media** Using the Device Stage **Manage media on your device** task a user can synchronize media files and playlists between the device and a computer running Windows 7, Windows Vista, or Windows XP.

These basic capabilities meet the minimum functionality requirements for Windows logo certification in the Device Stage portable media player (PMP) and other portable devices (OPD) categories. For more information about Device Stage categories, see the section [Windows Logo Certification](#). Your device will automatically create a baseline presentation when you include MTP Responder catalog items in your OS design, configure registry settings with information about your device, and implement OAL support for battery and storage status.

The customized presentation extends the baseline presentation by adding features such as branding logos, additional presentation images, customized tasks for managing the device, and links to provide users with access to product registration, support, manuals, applications, and accessories. To create a customized presentation, first configure your device to support the baseline presentation, and then create a device metadata package for installation on the user's computer to add additional end-user features. You can also implement extra functionality in your device to support customized Device Stage tasks as part of your customized presentation.

## Windows Logo Certification

Windows classifies devices into categories based on the functionality of the device. The MTP functionality in MTP Responder for Windows Embedded Compact 7 meets Windows logo certification requirements for two device categories: OPD and PMP. The OPD category includes portable navigation devices, consumer Internet devices, digital picture frames, e-book readers, portable gaming devices, and set-top boxes. The OPD category is a member of the Portable Device class. The Portable Device class also contains categories for cellular phones, digital cameras, and portable media players. The PMP category, also a member of the Portable Device class, consists of devices that natively support at least one file format for audio or video playback.

Each device category has unique requirements that a device must meet to be compatible with Device Stage and to qualify for a Windows logo. The MTP components included in MTP Responder for Windows Embedded Compact 7 fulfill requirements only for the PMP and OPD categories. However,

you can extend the included MTP Responder source code to meet the requirements for other Windows device categories.

For more information about Windows logo certification and the requirements for device certification, see the [Windows Logo Program](http://go.microsoft.com/fwlink/?linkid=183526) (<http://go.microsoft.com/fwlink/?linkid=183526>).

## MTP Responder Components

MTP Responder is composed of several major functional blocks: the MTP responder stack, MTP storage, and MTP transports. You select MTP Responder functionality from the Catalog Items view in Microsoft Platform Builder, where several catalog items are presented: MTP Responder (default), MTP Responder (minimal), MTP USB Transport, and MTP IP Transport. For more information about catalog items, see the section [Step 2 Add MTP Responder Catalog Items to Your OS Design](#).

### MTP Responder Stack

The MTP responder stack provides MTP router, dispatcher, and command handler functionality to support Device Stage communication with the MTP initiator included in Windows 7. The MTP responder stack communicates with the host computer through one or more MTP transports. You can configure the MTP responder stack to use a USB connection, a TCP/IP connection, or both. You add MTP responder stack functionality by adding one of the MTP responder stack catalog items in the **Catalog Items** view of Platform Builder. To support Device Stage, you must include the MTP responder stack in your OS design.

### MTP Storage

The MTP Media Metadata DB (Media MDB) storage provides functionality with which users browse and manage files on the device and transfer files between the device and a computer. Media MDB storage is included in the MTP responder stack and supports file browsing and media file synchronization from the host computer. The Media MDB uses the Windows Embedded Compact 7 Media Library to store information about MTP objects on the device. This supports the minimum requirements for Windows logo certification for both the OPD and PMP categories. For more information about the Media Library, see [Media Library](http://go.microsoft.com/fwlink/?LinkId=228737) (<http://go.microsoft.com/fwlink/?LinkId=228737>) in Windows Embedded Compact 7 Documentation.

### MTP Transports

The MTP USB transport provides MTP connectivity between the device and the host computer through a USB connection. The MTP IP transport provides MTP connectivity between the device and a host computer through a TCP/IP connection. You can use one or both of these transports in your device, but you must enable at least one transport to support Device Stage. Users have the ability to perform the same file browsing and device management operations over a TCP/IP connection to the device as with a USB connection.

## Prerequisites

To build support for Device Stage into your device with MTP Responder for Windows Embedded Compact 7, you use the tools that you typically use for Windows Embedded Compact development. You use additional tools and software development kits to test your Device Stage presentation and to create and test device metadata packages.

## Windows Embedded Compact 7

Windows Embedded Compact 7 includes an MTP responder and related functionality that is designed for servicing Device Stage requests.

You use the Microsoft Visual Studio 2008 integrated development environment (IDE) with the Windows Embedded Compact Platform Builder toolset to design, create, build, test, and debug your Windows Embedded Compact-based run-time image. For more information about Windows Embedded Compact development and Platform Builder, see [Integrated Development Environment Build Process](http://go.microsoft.com/fwlink/?LinkId=178104) (<http://go.microsoft.com/fwlink/?LinkId=178104>).

## Metadata Tools

To support the Device Stage customized presentation, you must assemble the files that make up a Device Stage customized presentation into a device metadata package for installation on the Windows 7 host computer. The baseline presentation does not require a device metadata package. For more information about creating a device metadata package for a customized device presentation, download the [Microsoft Device Experience Development Kit](http://go.microsoft.com/fwlink/?LinkId=178109) (<http://go.microsoft.com/fwlink/?LinkId=178109>).

## Windows 7

Windows 7 provides an MTP initiator for testing your device in addition to the environment for building and testing device metadata packages.

## Development Tasks

To configure your Windows Embedded Compact OS design to support Device Stage, complete the following tasks.

1. Create a baseline Device Stage presentation, which displays some identifying information about your device, your device icon, and a standard set of status elements and tasks.
  - a. Choose and include MTP Responder catalog items in your OS design.
  - b. Populate device registry keys and configuration files with information about your device.
  - c. Develop your OAL to support Device Stage status and task elements.
  - d. Handle MTP Responder notifications about changes in connection state.
  - e. Create a device icon image that represents your device.

2. Optionally, create a customized Device Stage presentation, which includes a user interface for your device that installs on Windows 7.
  - a. Create a device metadata package that installs on the host computer.
  - b. Develop any customized tasks in your device to support your device metadata package.
  - c. Submit your device for Windows logo certification.
  - d. Submit your device metadata package to Microsoft for signing.

When you complete task 1, Device Stage automatically generates a baseline presentation for your device. If you include the Media MDB storage option in your catalog item selection, this baseline presentation will include support for media file synchronization. When you complete tasks 1 through 2, Device Stage displays your device with a customized presentation. Windows logo certification and metadata signing are required only for a customized presentation.

If you create a customized presentation, you can add status displays and tasks that are specific to your device, promote your brand by adding customized graphic elements, and provide users with links to product enhancements, registration sites, support sites, and product manuals. If you don't create a user interface, Device Stage shows the baseline presentation for your device.

For host computers running Windows Vista or Windows XP, completing task 1 provides full MTP connectivity between your device and the host computer. Completing task 1 also provides media synchronization and file transfer capabilities for your device. The following steps and the links to related web sites, software development kits, and additional documentation will help you to prepare your device for Device Stage.

## Device Stage Implementation

---

To add Device Stage support to your device by using MTP Responder for Windows Embedded Compact 7, follow these steps:

### Step 1 Verify System Requirements

MTP Responder for Windows Embedded Compact 7 requires the following minimum system configurations.

#### Windows 7 and Windows Vista

If your development computer runs on Windows 7 or Windows Vista, the following hardware and software is required.

#### Hardware Requirements

- Computer with a 1 GHz or faster processor
- 1 GB of RAM



- 18 GB of available space on the installation drive
- 1 GB of available space on the system drive
- DVD-ROM drive
- Monitor that supports at least 1,024 × 768 screen resolution with 16-bit color

### Software Requirements

- Visual Studio 2008
- Windows Embedded Compact 7

## Windows XP

If your development computer runs on 32-bit Windows XP with Service Pack 3, the following hardware and software is required.

### Hardware Requirements

- Computer with a 933 MHz or faster processor (2 GHz recommended)
- 512 MB of RAM (1 GB recommended)
- 18 GB of available space on the installation drive
- 1 GB of available space on the system drive
- DVD-ROM drive
- Monitor that supports at least 1,024 × 768 screen resolution with 16-bit color

### Software Requirements

- Visual Studio 2008
- Windows Embedded Compact 7

When you install Windows Embedded Compact 7 and click **Installs all required, optional, and 3rd party options** in the **Install Options** dialog box, all the functionality required for MTP Responder development is automatically included. If you click **Customize your installation** in the **Install Options** dialog box, select the Platform Builder option in the **Customize Installation** dialog box. For more information about development with Platform Builder, see [Integrated Development Environment Build Process](http://go.microsoft.com/fwlink/?LinkId=178104) (<http://go.microsoft.com/fwlink/?LinkId=178104>).

## Step 2 Add MTP Responder Catalog Items to Your OS Design

To add MTP Responder support to your OS design, add one or more of the following catalog items and SYSGEN variables to your OS design. Use the MTP Responder (default) catalog item unless you intend to modify or replace the provided transport components. If you do not use MTP Responder

(default), you must include MTP Responder (minimal) and at least one transport, or use equivalent components of your own design.

**Table 1 - SYSGENs**

Catalog item	SYSGEN variable	Description
MTP Responder (default)	SYSGEN_MTP_RESPONDER	Adds the following components: MTP responder stack, Media MDB storage, and MTP USB transport
MTP Responder (minimal)	SYSGEN_MTP_RESPONDER_MIN	Adds MTP responder stack and Media MDB storage but no MTP transports.
MTP USB Transport	SYSGEN_MTP_RESPONDER_USB	Adds MTP over USB transport functionality.
MTP IP Transport	SYSGEN_MTP_RESPONDER_IP	Adds MTP over IP transport functionality.

When you select MTP Responder (default) or the MTP Responder (minimal), you implicitly include the Windows Embedded Compact Media Library catalog item. When you include the Media Library, you must explicitly enable support for specific audio and video formats by adding any associated codecs, file support, filters, or renderers that you will use with MTP Responder. For more information, see [Encoded Media Catalog Items and Sysgen Variables](http://go.microsoft.com/fwlink/?LinkId=228802) (http://go.microsoft.com/fwlink/?LinkId=228802).

For information about selecting catalog items, see [Adding Catalog Items to an OS Design](http://go.microsoft.com/fwlink/?LinkId=179330) (http://go.microsoft.com/fwlink/?LinkId=179330). For information about setting SYSGEN variables, see [Setting or Clearing a Sysgen Variable](http://go.microsoft.com/fwlink/?LinkId=179331) (http://go.microsoft.com/fwlink/?LinkId=179331).

## Step 3 Configure Registry Settings in Your OS Design

To configure registry settings in your OS design to support Device Stage, modify the mediaapps.reg file, which can be found at this location:

%\_WINCEROOT%\Public\Mediaapps\Oak\Files\Mediaapps.reg

This file contains a set of registry keys that you use to configure MTP Responder.

## Device Information

To identify and install the device in Windows, associate the device with a metadata package on the computer, and display information about the device in a Device Stage baseline presentation, you must add the device information entries to the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder**

These entries are listed in the following table.

#### Device Information Registry Settings

Name	Type	Default value	Description
DeviceModelName	REG_SZ	Windows Embedded Generic Device	The device model name that MTP uses to respond to GetDeviceInfo requests, for example, "Contoso GPS."
DeviceFriendlyName	REG_SZ	Windows Embedded Generic Device	The friendly, or display name that MTP uses to respond to GetDevicePropDesc requests, for example "Contoso GPS."
ModelID	REG_SZ	Generated if not specified	<p>A 128-bit GUID that associates a device with a Device Stage presentation, regardless of how the device is connected to the computer. For example, "{52620BB4-2F18-4e92-9494-A03A38719}."</p> <p>For more information about hardware IDs and model IDs, see the following documents in the <a href="http://go.microsoft.com/fwlink/?LinkID=178109">Microsoft Device Experience Development Kit</a> (<a href="http://go.microsoft.com/fwlink/?LinkID=178109">http://go.microsoft.com/fwlink/?LinkID=178109</a>).</p> <ul style="list-style-type: none"> <li>Windows 7 Device Stage Portable Device Class Development Guide</li> <li>Windows 7 Device Stage Reference Guide</li> </ul> <p>See also <a href="http://go.microsoft.com/fwlink/?LinkID=179337">ModelID Element</a> (<a href="http://go.microsoft.com/fwlink/?LinkID=179337">http://go.microsoft.com/fwlink/?LinkID=179337</a>).</p> <p>If you do not specify the ModelID, MTP Responder will generate one for you.</p>
DeviceVersion	REG_SZ	No default	The version information that MTP uses to respond to GetDeviceInfo requests, for example "1.2."
FunctionalID	REG_SZ	Generated if not specified	<p>The 128-bit GUID, permanent for the life of the device, that uniquely identifies an MTP device that is connected through multiple transports, for example "{2BB4074E-E469-4d1d-A028-615D91AA4D21}."</p> <p>For more information about the functional ID, see the <a href="#">MTP Device Services Extension</a></p>

Name	Type	Default value	Description
			<a href="#">Specification</a> (http://go.microsoft.com/fwlink/?LinkID=178887). If you do not specify the FunctionalID, MTP Responder will generate one for you at device boot time.
ContainerID	REG_SZ	Generated if not specified	An identifier that informs Windows 7 that multiple functional device instances actually originate from the same physical device. For more information about container IDs, see <a href="#">Multifunction Device Support and Device Container Groupings</a> (http://go.microsoft.com/fwlink/?LinkID=183536). If you do not specify the ContainerID, MTP Responder will generate one for you at device boot time.

**Note**

Device Stage displays your device as a “Composite (multi transport) Device” even if the device has only one MTP transport configured. MTP Responder reports a functional ID to the MTP initiator on the user’s computer, even if you do not configure FunctionalID in the registry settings previously mentioned. Therefore, Device Stage assumes that your device supports MTP connections over more than one transport. For more about multi transport devices, see [Multi-Transport Devices in Windows 7](#) (http://go.microsoft.com/fwlink/?LinkID=179542).

To disable the FunctionalID or ModelID settings, modify the Devicesettings.xml file, which can be found at this location:

%\_WINCEROOT%\Public\Mediaapps\Oak\Files\Devicesettings.xml.

In <DevicePropertiesSupported>, find the declarations for the FunctionalID and ModelID properties.

```
<DevicePropertiesSupported>
  <Base>0x5001</Base> <!-- BATTERYLEVEL -->
  <Base>0xD301</Base> <!-- FUNCTIONID -->
  <Base>0xD302</Base> <!-- MODELID -->
  <Base>0xD401</Base> <!-- SYNCHRONIZATIONPARTNER -->
  <Base>0xD402</Base> <!-- DEVICEFRIENDLYNAME -->
  <Base>0xD405</Base> <!-- DEVICEICON -->
</DevicePropertiesSupported>
```

When you remove the **FUNCTIONID** element previously shown, MTP Responder will not report a functional ID to the MTP initiator, even if you specify the FunctionalID in the registry. When you remove the **MODELID** element previously shown, MTP Responder will not report a model ID to the MTP initiator, even if you specify the ModelID in the registry.

## Transport Information

If you are using the MTP USB transport, you must add the product ID and vendor ID settings to the registry key:

**HKEY\_LOCAL\_MACHINE\Drivers\USB\FunctionDrivers\MTPUSBFn**

These settings are described in the following table.

### USB Transport Registry Settings

Name	Type	Default value	Description
idProduct	REG_DWORD	0x0622	A bus-specific product identifier that associates a device with a Device Stage presentation, for example, DWORD:0622. For more information, see <a href="http://go.microsoft.com/fwlink/?LinkId=179334">USB Function Client Driver Registry Settings</a> (http://go.microsoft.com/fwlink/?LinkId=179334).
idVendor	REG_DWORD	0x045E	<p>A vendor ID is the four-digit vendor code that the USB committee assigns to the vendor. Remember to change this value to your vendor ID; 045E is reserved for use by Microsoft. For more information, see:</p> <ul style="list-style-type: none"> <li>• <a href="http://go.microsoft.com/fwlink/?LinkId=183535">Device Identification String</a> (http://go.microsoft.com/fwlink/?LinkId=183535)</li> <li>• <a href="http://go.microsoft.com/fwlink/?LinkId=179341">Identifiers for USB Devices</a> (http://go.microsoft.com/fwlink/?LinkId=179341)</li> <li>• <a href="http://go.microsoft.com/fwlink/?LinkId=179334">USB Function Client Driver Registry Settings</a> (http://go.microsoft.com/fwlink/?LinkId=179334)</li> </ul> <p>To obtain a vendor ID, see <a href="http://go.microsoft.com/fwlink/?LinkId=183531">Universal Serial Bus</a> (http://go.microsoft.com/fwlink/?LinkId=183531). Also see <a href="#">Standard USB Identifiers</a></p>

Name	Type	Default value	Description
			( <a href="http://go.microsoft.com/fwlink/?LinkID=179342">http://go.microsoft.com/fwlink/?LinkID=179342</a> ).

If you are using the MTP IP transport, you must add the product description and manufacturer URL to the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder**

These settings are described in the following table.

#### IP Transport Registry Settings

Name	Type	Default value	Description
ProductDescription	REG_SZ	Empty String/NULL	The description of the device that is displayed. For example, "Portable Navigation Device with 4.3-inch screen."
ManufacturerURL	REG_SZ	Empty String/NULL	The URL for your company's web site.

## Device Metadata Service

To transfer a customized presentation from the device to the computer when the device first connects to the MTP initiator, you must add the Path, ContentID, and Flags registry settings to the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder\Metadata\<Locale>**

where <Locale> is the value from the **Locale** element in PackageInfo.xml in the device metadata package. These settings are described in the following table.

#### Metadata Service Registry Settings

Name	Type	Default value	Description
Path	REG_SZ	No default	The absolute path (on the device) to the device metadata package for the locale, for example Hard Disk1\WDS Metadata.
ContentID	REG_SZ	No default	A unique GUID string in the registry in CLSID format (

Name	Type	Default value	Description
			"{1234-...}" ) that is assigned by the Windows logo signing process, for example "{C23954F9-80A1-497d-AB9A-EFE0EFCEAA9C}."
Flags	REG_DWORD	No default	Set to 1 to designate that this package has the attribute <locale default="true"> in the PackageInfo.xml file in a device metadata package; otherwise, 0 (zero).

For a list of locale identifier strings, see [Locale Identifier Constants and Strings](http://go.microsoft.com/fwlink/?LinkID=183530) (http://go.microsoft.com/fwlink/?LinkID=183530).

For more information about how Windows 7 uses the locales specified in device metadata packages, see "How to Localize Device Stage Experiences" in the Windows 7 Device Stage Reference Guide in the [Microsoft Device Experience Development Kit](http://go.microsoft.com/fwlink/?LinkID=178109) (http://go.microsoft.com/fwlink/?LinkID=178109).

## MTP Storage Settings

For users to be able to browse files on their device from a computer you must define the location for storage of media files and MTP objects on the device. You define the storage location by configuring a registry key in mediaapps.reg.

### Storage Location

For the storage location on your device, you create a registry key

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder\Storage\<Name>**

where <Name> is the unique name that you choose to label that storage location.

For example, if your device has internal flash memory, you could define the following registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder\Storage\Internal**

In the registry key for the storage location, you configure the StorageRoot, StorageDescription, and VolumeIdentifier settings, as listed in the following table.

### MTP Storage Location Settings

Name	Type	Default value	Description
StorageRoot	REG_SZ	None	The path to this MTP

Name	Type	Default value	Description
			storage location. If you do not specify a StorageRoot path, MTP Responder uses the default storage described in the section "Default Storage."
StorageDescription	REG_SZ	"MTP Storage"	A human-readable string, for example, "Internal Flash Memory."
VolumeIdentifier	REG_SZ	"MTP Volume Identifier"	A unique identifier, such as a serial number. Only the first 128 characters are used to identify the storage device, and they must be unique for the storage device.

## Default Storage

When an MTP initiator transfers a media file to MTP Responder without specifying a storage location on the device, MTP Responder stores this file in the default storage location. You can indicate the storage location MTP Responder will use as the default storage location. To set the default storage location, you configure the DefaultStorage setting in the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder\Storage**

### Default Storage Setting

Name	Type	Default value	Description
DefaultStorage	REG_SZ	None	The name of the default MTP storage location. This name must be identical to the registry key name that you used for this storage location.  If you do not include a name here, MTP



Name	Type	Default value	Description
			Responder uses the first enumerated storage. For more information, see the section “Storage Location.”

If you do not specify a value for the default storage location and you have not specified a storage location, MTP Responder will use the “My Documents” folder. If the “My Documents” folder is not available, MTP Responder uses “\MTPStorageRoot.”

Using the names shown in the section “Storage Location,” you can define the following mount point:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder\Storage\Internal**

using “\OnBoardFlash\MediaPartitionRoot” as the value of StorageRoot.

You then configure:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder\Storage**

using the value “Internal” for DefaultStorage.

## Media MDB Settings

To use the Media MDB storage, you must also configure a thumbnail storage location.

### Note

Do not configure a watch location that monitors folders or subdirectories in any part of the MTP storage. If a watch location has been defined, the storage that is defined as watch location will not be synced from the host computer.

## Thumbnail Storage

With Media MDB storage, your device automatically supports storage and synchronization of thumbnails sent from the MTP initiator. Media MDB thumbnails, also referred to as “representative samples” in the MTP 1.0 Specification, can be small versions of album art for videos, photos, and albums that support them. You must configure the thumbnail storage location, size range, height range, and width range.

To configure the thumbnail storage location, specify the ThumbnailRoot setting in the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Thumbnail**

This setting is described in the following table.

**Thumbnail Storage Setting**

Name	Type	Default value	Description
ThumbnailRoot	REG_SZ	\ThumbnailRoot	The path to the location where thumbnails are to be stored.

If you do not specify the default value, MTP Responder will default to “\ThumbnailRoot.”

To configure the thumbnail size range, specify the MinimumSize, MaximumSize, and StepSize settings in the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Thumbnail\Size**

These settings are described in the following table.

**Thumbnail Size Range Setting**

Name	Type	Default value	Description
MinimumSize	REG_DWORD	0	The minimum size of the thumbnail.
MaximumSize	REG_DWORD	1200000 (1.2MB)	The maximum size of the thumbnail
StepSize	REG_DWORD	1	The step size between minimum and maximum size of the thumbnail.

To configure the thumbnail height range, specify the MinimumHeight, MaximumHeight, and StepSize settings in the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Thumbnail\Height**

These settings are described in the following table.

**Thumbnail Height Range Setting**

Name	Type	Default value	Description
MinimumHeight	REG_DWORD	32	The minimum height of the thumbnail.
MaximumHeight	REG_DWORD	192	The maximum height of the thumbnail.

Name	Type	Default value	Description
StepSize	REG_DWORD	16	The step size between minimum and maximum height of the thumbnail.

To configure the thumbnail width range, specify the MinimumWidth, MaximumWidth, and StepSize settings in the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Thumbnail\Width**

These settings are described in the following table.

#### Thumbnail Width Range Setting

Name	Type	Default value	Description
MinimumWidth	REG_DWORD	32	The minimum width of the thumbnail.
MaximumWidth	REG_DWORD	192	The maximum width of the thumbnail.
StepSize	REG_DWORD	16	The step size between minimum and maximum width of the thumbnail.

The Media Library automatically generates thumbnails when it parses incoming images. You can configure the parameters for thumbnail generation, including thumbnail size and thumbnail file format. In addition, the Media Library provides a registry location that you can change to disable automatic thumbnail generation. For more information, see [Media Library](http://go.microsoft.com/fwlink/?LinkId=228737) (<http://go.microsoft.com/fwlink/?LinkId=228737>).

## Step 4 Provide OAL Support for Device Stage

When a computer issues an MTP GetDeviceInfo request to your device, MTP Responder gets information about the device from the [SystemParametersInfo](http://go.microsoft.com/fwlink/?LinkId=179347) (<http://go.microsoft.com/fwlink/?LinkId=179347>) function.

To set that information, issue adaptation layer I/O control codes (IOCTLs) to inform the kernel of information about your device by calling the [OEMIoControl](http://go.microsoft.com/fwlink/?LinkId=228820) (<http://go.microsoft.com/fwlink/?LinkId=228820>) function on device startup. You use the IOCTLs described later in this article to set the manufacturer name and the device ID, and to add support for the battery level and free storage displays.

## Manufacturer Name

To set the manufacturer name use the following function call, where *lpOutBuf* is a pointer to a buffer that contains the string that represents your manufacturer name, and *lpBytesReturned* is the number of bytes that you wrote to *lpOutBuf*.

```
OEMIoControl(
    IOCTL_HAL_GET_DEVICE_INFO,
    SPI_GETPLATFORMMANUFACTURER,
    4,
    lpOutBuf,
    lpBytesReturned)
```

## Device ID

To set the device ID, use the following function call, where *lpOutBuf* is a pointer to a buffer that contains the string that represents the device ID, and *lpBytesReturned* is the number of bytes that you wrote to *lpOutBuf*.

```
OEMIoControl(
    IOCTL_HAL_GET_DEVICE_INFO,
    SPI_GETUUID,
    4,
    lpOutBuf,
    lpBytesReturned)
```

For more information about the device ID, see [Device ID](http://go.microsoft.com/fwlink/?LinkId=228825) (<http://go.microsoft.com/fwlink/?LinkId=228825>) and [How do I get the "right" Device ID?](http://go.microsoft.com/fwlink/?LinkId=179345) (<http://go.microsoft.com/fwlink/?LinkId=179345>).

## Battery Level

To add support for battery level, implement the **IOCTL\_BATTERY\_GETSYSTEMPOWERSTATUSEX2** control in your battery driver code. This control returns a **SYSTEM\_POWER\_STATUS\_EX2** structure that reports the battery level in the `BatteryLifePercent` field. Battery level is expressed as a percentage, with a value ranging from 0 to 100. The value that your battery driver code writes into `BatteryLifePercent` shows up as the battery level status for your device in the **Devices and Printers** folder on the computer. You must also enable **SYSGEN\_BATTERY** to support battery level status in your device.

Device Stage on the desktop computer issues an MTP GetDevicePropDesc request to MTP Responder to retrieve the battery level on your device. MTP Responder, in turn, calls the **GetSystemPowerStatusEx2** function to determine the current battery level to send back to Device Stage.

You must implement support for battery level status even if the device has no battery. For example, if the device must be connected to wall current to operate, you still must support the **IOCTL\_BATTERY\_GETSYSTEMPOWERSTATUSEX2** control in your OAL, return a **SYSTEM\_POWER\_STATUS\_EX2** structure, and report a power level of 100 percent.

For more information, see [IOCTL\\_BATTERY\\_GETSYSTEMPOWERSTATUSEX2](http://go.microsoft.com/fwlink/?LinkId=228828) (<http://go.microsoft.com/fwlink/?LinkId=228828>). For more information about **GetSystemPowerStatusEx2**, see [GetSystemPowerStatusEx2](http://go.microsoft.com/fwlink/?LinkId=179350) (<http://go.microsoft.com/fwlink/?LinkId=179350>).

## Free Storage Space

Add support for free storage space status if you are implementing a device file system for use with MTP Responder. To add support for free storage space status, ensure that **GetDiskFreeSpaceEx** properly returns the total capacity and the amount of free storage on your device.

Device Stage on the desktop computer issues an MTP GetStorageInfo request to MTP Responder to retrieve information about physical storage on your device. In turn, MTP Responder calls the **GetDiskFreeSpaceEx** function to determine the total capacity and the amount of free space to send back to Device Stage. This function returns information about the amount of space on the local file system of the device. **GetDiskFreeSpaceEx** returns the total amount of space and the total amount of free space; these quantities appear next to the image for your device in Device Stage.

For more information, see [GetDiskFreeSpaceEx](http://go.microsoft.com/fwlink/?LinkId=179351) (<http://go.microsoft.com/fwlink/?LinkId=179351>).

## File Time Stamps

To meet the minimum requirements for Windows logo certification, your device file system must support the ability to set the date and time that a file was created, last accessed, or last modified. For more information, see [SetFileTime](http://go.microsoft.com/fwlink/?LinkId=193263) (<http://go.microsoft.com/fwlink/?LinkId=193263>).

## Step 5 Optionally Handle Session Notifications

MTP Responder supports notifications to inform your device about the status of the connection with the MTP initiator. You can use notifications to help your device react to changes in MTP session state and to authenticate incoming TCP/IP connections from MTP initiators.

When an MTP responder interacts with an MTP initiator, it must have exclusive access to the media so that all content can be properly synchronized with the MTP initiator. Therefore, you must coordinate content access between the MTP responder and other applications on your device.

When MTP Responder connects to an MTP initiator, it generates a notification to inform your device that an MTP session has started. When your device receives this notification, it should do the following:

- Pause user interactions with any applications that access media files to prevent unsynchronized changes in this content.
- Ensure that all content file handles are unlocked and closed.

When the MTP session ends, MTP Responder generates a notification to indicate that the MTP connection is no longer active.

To register for MTP session notifications, you create named event objects for each event type and store the name of each event object in the MTP Responder registry location for that notification type. You can create a notification thread that waits on these event objects and then reacts to the change in MTP session state when the thread becomes active.

Specifically, for MTP session notifications, create two named event objects and store these names in the SessionActiveNamedEvent and SessionInactiveNamedEvent settings in the registry key:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\MTP\Responder**

These settings are described in the following table.

### Session Event Settings

Name	Type	Default value	Description
SessionActiveNamedEvent	REG_SZ	L"MTP_SESSION_ACTIVE"	The name of the event object to be notified when MTP Responder connects to an MTP initiator.
SessionInactiveNamedEvent	REG_SZ	L"MTP_SESSION_INACTIVE"	The name of the event object to be notified when MTP Responder disconnects from the MTP initiator.

An example of a sequence of steps that will handle session notifications is as follows:

1. Populate the registry using the keys shown in the preceding table for the session active and session inactive events or use the default session named events.
2. Create an application to perform steps 3 through 10.
3. Read the registry values for the two events.
4. Call [CreateEvent](http://go.microsoft.com/fwlink/?LinkID=210951) (http://go.microsoft.com/fwlink/?LinkID=210951) to create events for MTP Responder connect (session active) and disconnect (session inactive), and an event for notification thread shutdown.

5. Use [CreateThread](http://go.microsoft.com/fwlink/?LinkID=227552) (<http://go.microsoft.com/fwlink/?LinkID=227552>) to create a notification thread that executes in a continuous loop, activates the shutdown event when a shutdown request (like pressing a key) is received, and then exits.
6. Call [WaitForMultipleObjects](http://go.microsoft.com/fwlink/?LinkID=228859) (<http://go.microsoft.com/fwlink/?LinkID=228859>) to wait on the shutdown event and the session active event. When MTP Responder connects, it activates the session active event and the event remains activated for the duration of the session, so that your application will receive the event notification even if it starts after the event is activated.
7. Perform the actions needed for session activation.
8. Call [WaitForMultipleObjects](http://go.microsoft.com/fwlink/?LinkID=228859) (<http://go.microsoft.com/fwlink/?LinkID=228859>) to wait on the shutdown event and the session inactive event. When MTP Responder disconnects, it activates the session inactive event and the event remains activated. Your application will wait on the session active event if it starts after the inactive event is activated.
9. Perform the actions needed for session inactivation.
10. Repeat from step 6.

## Step 6 Optionally Handle MTP IP Authentication

MTP IP Authentication manages incoming connections from MTP initiators over TCP/IP networks. You can optionally use these functions to allow or reject a pending MTP IP connection by registering for notifications of incoming connections. You can use this notification to verify that a pending connection originates from a valid MTP initiator.

For more information about MTP IP Authentication, see [MTP IP Authentication Functions](http://go.microsoft.com/fwlink/?LinkID=228860) (<http://go.microsoft.com/fwlink/?LinkID=228860>).

## Step 7 Optionally Handle Power Management Scenarios

To avoid file system or database corruption in situations when power is partially or completely lost, you must notify the MTP transport service to halt so that it does not attempt to write any data. You are responsible for handling all power situations; MTP does not handle any power scenarios for you. These scenarios may include:

- Bringing down the MTP stack in a low-power or power-loss situation to prevent data corruption during file synchronization.
- Bringing the MTP stack back online after the device has been plugged into a power source to allow synchronization without a device restart.

To handle these situations, you must create a power notification handler service that will receive low-power, shutdown, and possibly reset notifications. When it receives a power notification, your service must stop the MTP transport so that no data is written. To do this, your service will send IOCTLs to the MTP transport service. For more information, see [IOCTLs that are Sent by Applications](http://go.microsoft.com/fwlink/?LinkID=199090) (<http://go.microsoft.com/fwlink/?LinkID=199090>).

The shutdown notification will require registering your service with the power manager so that it can be notified about power changes. On shutdown, your service will receive an **IOCTL\_POWER\_SET** call with power state set to indicate shutdown. This is a blocking call, so your service can stop the MTP transport service and then return. After your service returns, the shutdown sequence will continue.

## Step 8 Create a Presentation for Windows 7

You can use either the baseline presentation for Device Stage, which is the default, or you can create a customized presentation.

### Device Icon

When your device connects for the first time to a computer running Windows 7, the computer attempts to load an icon file from your device, and then to store this icon in a Device Manager node on the computer. A device icon is recommended, but not required, for a baseline presentation for devices in the portable devices category. For customized presentations, the device icon is required in the metadata package, but it is not required on the device.

#### To add an icon to your device

1. Create an icon file in the .ico file format. See the instructions for creating icons in the Windows 7 Device Stage Design Guide, which is in the [Microsoft Device Experience Development Kit](http://go.microsoft.com/fwlink/?LinkID=178109) (<http://go.microsoft.com/fwlink/?LinkID=178109>).
2. Name your device icon MTPdeviceicon.ico.
3. Copy MTPdeviceicon.ico to the \Windows directory during binary image build. For more information about binary image build, see [Binary Image Builder File](http://go.microsoft.com/fwlink/?LinkID=179354) (<http://go.microsoft.com/fwlink/?LinkID=179354>).

### Baseline Presentation

The baseline presentation is a good choice if you do not want to develop a customized metadata package and submit it to Microsoft to be signed. You can replace the baseline presentation later with a customized presentation.

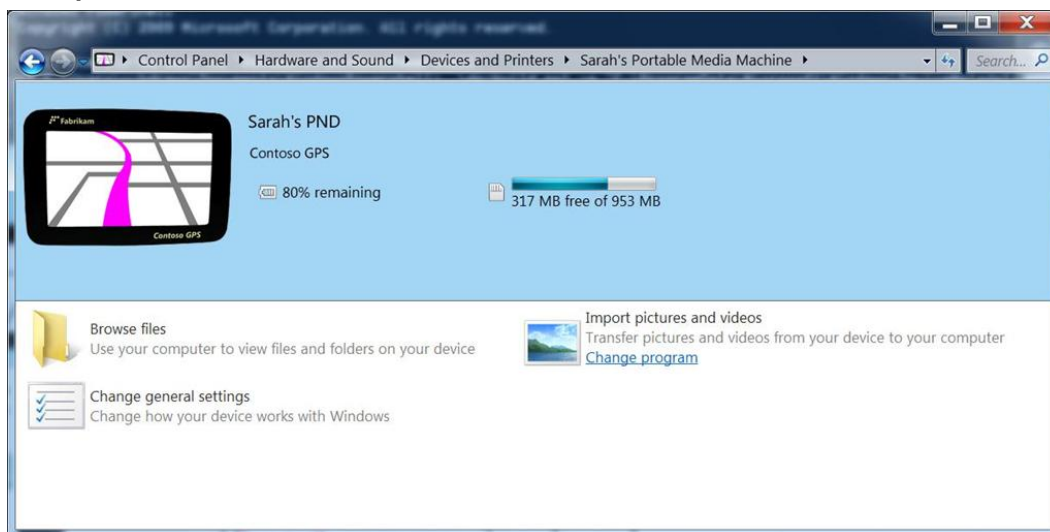
The baseline presentation contains the following:

- Status information from device drivers, such as battery level, storage space, and storage configuration.
- Information from the registry on your device, such as device display name and model and manufacturer name.
- A device icon from the \Windows directory on the device.

The following figure shows a baseline presentation in the Device Stage window.



## Example Baseline Presentation



For more information about the Device Stage baseline presentation, see Windows 7 Device Stage Portable Devices Development Kit in the [Microsoft Device Experience Development Kit](http://go.microsoft.com/fwlink/?LinkID=178109) (<http://go.microsoft.com/fwlink/?LinkID=178109>).

## Remove the Media Synchronization Task

The default Device Stage baseline presentation includes a task for media synchronization. If you do not want to support media synchronization, you can remove the media synchronization task so that it is not presented to the user.

You can remove the media synchronization task from the baseline presentation by installing a non-media baseline metadata package on your device. This package contains a version of the baseline presentation without a media synchronization task. The computer loads this metadata package from your device and uses it to display this new baseline presentation instead of the default baseline presentation.

The non-media baseline metadata package is available in the following location:

%\_WINCEROOT%\Public\Mediaapps\Oak\Files\5d32b836-5ed8-4454-b7b2-e885bf1b7b80.devicemetadata-ms.

### ▶ To install the non-media baseline metadata package on your device, do the following:

1. Set the Model ID for your device to the GUID value {4E794ABF-01F1-42c8-92C5-97C8CCB4AE41}. For more about Model ID, see [Device Information](#).
2. Include the non-media baseline metadata package in your binary image build (see [Binary Image Builder File](#) (<http://go.microsoft.com/fwlink/?LinkID=179354>)), and configure the device metadata service to transfer this metadata package from the device to the computer, when the device first connects to the computer. For more information about the device metadata service,

see [Device Metadata Service](#).

When the user's computer connects to your device, MTP Responder automatically copies the non-media baseline metadata package to the computer, and Device Stage displays the baseline presentation to the user without a media synchronization task.

## Custom Presentation

Customized presentations can include your company's branding elements and customized tasks that are specific to your device. When you create a customized presentation for your device, you must submit your device to Microsoft for Windows logo certification. After your device is certified, you can submit your metadata packages to Microsoft to be signed. Device Stage only displays signed device metadata packages.

A custom presentation contains the following:

- Status information, which is the same as in the baseline presentation, unless you add customized status information.
- Information from XML files in a device metadata package that you create for your device.

The following figure shows a custom presentation in the Device Stage window that includes branding elements and customized tasks.

### Example Custom Presentation



If you previously created a Device Stage metadata package for Windows Embedded CE 6.0, you can use that metadata package in Windows Embedded Compact 7 without modifications, or you can enhance it to take advantage of new features in Windows Embedded Compact 7. For information about creating metadata packages that drive customized Device Stage presentations, see the [Microsoft Device Experience Development Kit](http://go.microsoft.com/fwlink/?LinkID=178109) (<http://go.microsoft.com/fwlink/?LinkID=178109>).

## Step 9 Test Your Presentation

You can test your presentation on a computer running Windows 7.

### Baseline Presentation

Use the following steps to verify that your baseline presentation functions as expected.

#### To test a baseline presentation

1. Connect your device to the computer using a USB cable.
2. If Device Stage does not start, click **Start**, and then click **Devices and Printers**.
3. In the **Devices and Printers** folder, identify the icon and text that represent your device.
4. Right-click the icon, and then click **Properties**.
5. On the **General** tab, verify that the displayed text is correct, and then click **Cancel**.
6. In the **Devices and Printers** folder, double-click the icon for your device.
7. In the Device Stage window, verify that the icon and text appear as you intended.
8. If your device supports media synchronization, verify that, by clicking the **Manage media on your device task** button, you can transfer media files to and from your device and keep these files properly synchronized.

### Custom Presentation

Device Stage only displays device metadata packages that have been signed by Microsoft. To test a customized metadata package, you must put Windows into a special test-signing mode.

#### To put Windows into test-signing mode

1. Click **Start**, and then type **cmd** (do not press Enter).
2. In the list of search results, right-click **cmd.exe**, and then click **Run as Administrator**.
3. In the command prompt window, type **Bcdedit -set testsigning ON**, and then press **Enter**.
4. Restart your computer.

The text “Test Mode Windows 7” appears on the desktop when Windows is in test-signing mode.

#### To test a custom presentation

1. Copy the device metadata package to  
C:\ProgramData\Microsoft\Windows\DeviceMetadataStore\<Locale>, where <Locale> is the value from the **Locale** element in PackageInfo.xml in the device metadata package.

 **Note**

ProgramData is a hidden folder.

2. Follow steps 1 through 8 for the baseline presentation, as previously stated.
3. Verify that all the graphic elements, status displays, and tasks that you added to the device metadata package appear.
4. Verify that the status elements display correct values for your device.
5. Verify that all the tasks function as you expect.
6. Test all the links in your presentation.

## Step 10 Complete Windows Logo Testing

You can have your device certified for the Windows logo if it is a portable media player (PMP) device or if it is in any of the following Device Stage other portable devices (OPD) categories.

- Portable navigation devices
- Consumer Internet devices
- Digital picture frames
- E-book readers
- Portable gaming devices
- Set-top boxes

To qualify for Windows logo certification, OPDs must support battery level status and free storage status, and provide functionality to browse files. PMP devices must also support media synchronization. In addition, you must meet requirements for devices in each category to pass Windows logo testing. For more information, see the [Windows Logo Program website](http://go.microsoft.com/fwlink/?LinkId=183526) (<http://go.microsoft.com/fwlink/?LinkId=183526>).

## Step 11 Submit Your Presentation to Windows Quality Online Services

See the [Windows Quality Online Services website](http://go.microsoft.com/fwlink/?LinkId=179356) (<http://go.microsoft.com/fwlink/?LinkId=179356>) for information on submitting your customized metadata package to Microsoft to be signed.

## Optimizations and Modifications

---

After you have incorporated MTP Responder into your device, you can optionally adjust MTP Responder resource and performance parameters or modify the MTP Responder implementation to better suit your needs. You can optimize MTP Responder to make the best tradeoff between lower CPU and memory usage versus improved performance, modify the provided source code to add customizations and extensions, add support for additional MTP object properties and commands, and

download additional software development kits and programming documentation for MTP and Device Stage development.

For more information about MTP Responder optimization and modifications, see [Inside MTP Responder](http://go.microsoft.com/fwlink/?LinkID=210328) (<http://go.microsoft.com/fwlink/?LinkID=210328>).

## Conclusion

---

Device Stage technology presents a graphical interface that makes it easy for users to find and use applications and services for their device when it is connected to a computer running Windows 7. MTP Responder for Windows Embedded Compact 7 provides the functionality for the device to communicate with Device Stage.

You can choose to implement a baseline presentation or a customized presentation for your device in Device Stage. The baseline presentation provides the most essential device features and is simple to set up, whereas the customized presentation offers additional device features, but it requires you to develop a device metadata package.

MTP Responder for Windows Embedded Compact 7 includes the MTP storage component called Media MDB. With Media MDB storage, MTP Responder prepares your device to meet Windows logo requirements for devices in the other portable devices (OPD) and portable media player (PMP) categories.

To incorporate MTP Responder into your device for Device Stage support, you include MTP Responder catalog items in your OS design, configure device registry settings, add OEM Adaptation Layer (OAL) support, create presentation elements, and test your device with Device Stage. You can optimize MTP Responder performance or modify the provided source code to add new functionality. Finally, you can submit your Device Stage-compatible device for Windows logo certification.

## Additional Resources

---

To learn more about MTP Responder, see [Inside MTP Responder](http://go.microsoft.com/fwlink/?LinkID=210328) (<http://go.microsoft.com/fwlink/?LinkID=210328>).

To learn more about Windows 7 Device Stage, see the following links.

- [Windows Team Blog – The Device Experience in Windows 7](http://go.microsoft.com/fwlink/?LinkID=179365) (<http://go.microsoft.com/fwlink/?LinkID=179365>)
- [Windows Device Experience](http://go.microsoft.com/fwlink/?LinkId=183519) (<http://go.microsoft.com/fwlink/?LinkId=183519>)
- [Microsoft Device Experience Development Kit](http://go.microsoft.com/fwlink/?LinkId=178109) (<http://go.microsoft.com/fwlink/?LinkId=178109>)
- [Windows Logo Program](http://go.microsoft.com/fwlink/?LinkId=183526) (<http://go.microsoft.com/fwlink/?LinkId=183526>)

To learn more about Windows Embedded, see the following link.

- [Windows Embedded website](http://go.microsoft.com/fwlink/?LinkId=183524) (<http://go.microsoft.com/fwlink/?LinkId=183524>)

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.