



Inside MTP Responder

Writers: Mark McLemore, Harold Drews

Technical Reviewer: Alex Bussmann

Published: January 2012

Applies To: Windows Embedded Compact 7

Abstract

This paper describes the internal implementation of MTP Responder for Windows Embedded Compact 7. MTP Responder implements Media Transfer Protocol (MTP), a technology for connecting a device to a Windows 7 computer to synchronize and transfer files. This paper covers:

- The design and implementation of the responder and storage components
- How to optimize performance
- How to customize and extend the source code
- Where to locate additional development kits and documentation

[MTP Responder Deployment Guide](http://go.microsoft.com/fwlink/?LinkID=210234) (<http://go.microsoft.com/fwlink/?LinkID=210234>) is a prerequisite for this paper.

Contents

Introduction	3
MTP Responder Overview.....	3
MTP Responder Stack	4
Source Code Modification	5
MTP Storage.....	6
Media MDB Storage.....	6
Media MDB Operations and Objects	7
Media MDB MTP Operations	7
Media MDB Object Formats.....	10
Media MDB Object Properties.....	11
Media MDB Implementation.....	15
MTP Media Library Plug-in	16
Media MDB Optimization	16
Specifying the Maximum Number of Entries.....	17
Source Code Modification	17
Programming References.....	17
Conclusion	18
Additional Resources	18

Introduction

MTP Responder for Windows Embedded Compact 7 uses Media Transfer Protocol (MTP) to connect an embedded device with a host computer running Windows 7, Windows Vista, or Windows XP so that device information and files can be transferred and synchronized between the host and the device. MTP Responder for Compact 7 is based on the MTP responder in the Windows 7 Portable Device Enabling Kit for MTP, version 7R2. This kit contains the source code for the reference implementations of an MTP initiator and an MTP responder. Microsoft derived MTP Responder from the reference MTP responder in this kit, and added functionality so that Windows Embedded Compact 7 devices support Microsoft Device Stage operations on host computers running Windows 7. Compact 7 devices also support Windows Media Player and Windows Explorer operations on hosts running Windows 7, Windows Vista, or Windows XP. This article assumes that you have read the [MTP Responder Deployment Guide](http://go.microsoft.com/fwlink/?LinkID=210234) (<http://go.microsoft.com/fwlink/?LinkID=210234>), that you have included MTP Responder catalog items in your OS design, and that you have configured MTP Responder registry settings in your device to support Windows Media Player and Device Stage.

MTP Responder on your device communicates with the desktop computer using a USB or TCP/IP connection. Your device uses MTP Responder to provide information about itself and to transfer and synchronize file contents from the desktop computer. You use components of MTP Responder to implement functionality for your device to meet the Windows logo requirements in the portable media player (PMP) and other portable devices (OPD) categories.

This article explains the internal design and configurable parameters of the major components of MTP Responder. The following sections provide guidance on how you can customize these components, optimize their operating performance, and implement enhancements for your device. This guide also provides information on the source code provided with MTP Responder and lists references and sources of additional information to help you understand how to modify it to fully support the features and functionality of your device.

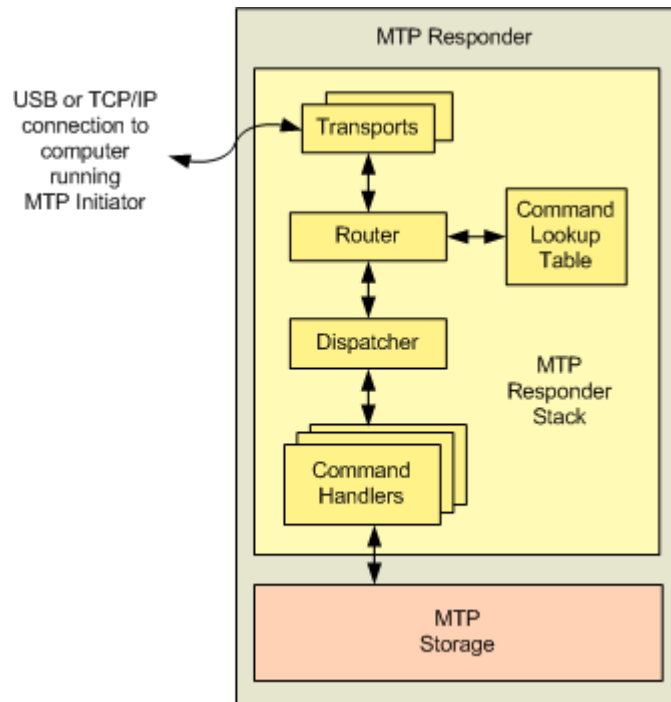
MTP Responder Overview

The following figure shows the two major functional blocks of MTP Responder: the MTP responder stack and MTP storage. The MTP responder stack implements the core MTP operations, events, properties, and object formats of the media transfer protocol described in the [MTP 1.0 Specification](http://go.microsoft.com/fwlink/?LinkID=183533) (<http://go.microsoft.com/fwlink/?LinkID=183533>).

The MTP responder stack also implements an extension to the MTP specification that provides support for MTP device services. Shown on the left side the figure, the MTP responder stack communicates with an MTP initiator through either a USB or a TCP/IP connection. For MTP operations that require access to files, metadata, or device properties, the MTP responder stack calls MTP storage (shown at the bottom of the figure) to process these requests. When you add the MTP Responder (default) or

MTP Responder (minimal) catalog items to your OS design, you include the MTP responder stack. MTP storage supports the MTP file formats, properties, and commands that are required for Windows Media Player and Device Stage compatibility. The Media Metadata Database (MDB) uses the Windows Embedded Compact 7 Media Library to store media file metadata across MTP sessions.

MTP Responder Components



When you add the MTP Responder (default) catalog item to your OS design, you automatically include MTP storage with the Media MDB database.

MTP Responder Stack

The MTP responder stack is made up of several subcomponents, as shown in the previous figure.

- **Transport** Controls data transport across a USB or TCP/IP connection with the MTP initiator. MTP Responder for Compact 7 includes two transports, one for USB and one for network connectivity.
- **Router** Routes MTP commands from each transport to the dispatcher, using the command lookup table to map each command to a route through the dispatcher and command handlers.
- **Command Lookup Table** Describes the route that an MTP command will take through the dispatcher and command handlers.

- **Dispatcher** Forwards MTP commands from the router to the appropriate command handler. In addition, if an MTP operation includes input data from the MTP initiator, the dispatcher routes the data for that operation to the command handler.
- **Command Handler** Implements the handler routines that process the various MTP operations supported by MTP Responder. Each command handler processes a particular MTP operation.

MTP Responder includes source code for a number of general purpose functions that are used extensively by the MTP responder stack. These general purpose functions are referred to collectively as the platform services layer.

For more information about the MTP responder stack and the platform services layer, see the [Media Transfer Protocol Porting Kit](http://go.microsoft.com/fwlink/?LinkID=183529) (<http://go.microsoft.com/fwlink/?LinkID=183529>).

Source Code Modification

You can modify or extend the provided MTP responder stack implementation to add Device Stage features that are specific to your device. For example, you can add MTP device services extensions to support device tasks for features such as credentials provisioning or device firmware upgrades. You can also create your own event handlers to report transfer progress or device connection and disconnection. In some cases, you may find it necessary to customize the MTP responder stack to make performance or resource optimizations unique to your device, or to add support for a new MTP transport protocol. You may need to make other changes to accommodate specific device communication with Windows Media Player on host systems that are running Windows 7, Windows Vista, or Windows XP. To make customized modifications and extensions to the MTP responder stack for your device, see the MTP Responder common source code at the following location:

%_WINCEROOT%\Public\Mediaapps\Oak\MTP\Common\MTP

These source files are derived from the MTP Responder reference implementation that is included in the Portable Device Enabling Kit. This implementation is not specific to Compact 7. These files include the core MTP responder stack functionality, which includes the following:

- MTP dispatcher
- MTP router
- MTP common command handler functionality
- MTP core command handler
- MTP command lookup table
- Common code to support MTP transports
- Support for USB, TCP/IP, and other transports
- Message queue library
- MTP format and properties library
- MTP event manager and event handlers
- MTP performance logger

In addition, MTP responder stack source code that is specific to Compact 7 is available at the following location:

%_WINCEROOT%\Public\Mediaapps\Oak\MTP\Wince\Src\MTP

These source files implement MTP extensions that integrate the MTP responder stack with Compact 7 in the following ways:

- Functionality to report media and Device Stage information back to the MTP initiator, such as device product description, manufacturer name, serial number, model ID, device icon, and battery level.
- Platform service layer functionality that is unique to Windows Embedded Compact 7, including atomic functions, debug utilities, memory management, thread management, timer utilities, sockets, and message queues.
- Implementation of the MTP status service: generation of reports about battery level and storage space by calling OEM adaptation layer (OAL) functions (such as **GetDiskFreeSpaceEx**), and returning the results to the MTP initiator.
- Implementation of the MTP device metadata service.
- MTP transport functionality for Windows Embedded Compact USB and network devices.

The MTP Responder source code is written in ANSI C. Depending on your objectives and the extent of the modifications or extensions that you want to make to the MTP Responder, you can modify any or all of these source code locations.

MTP Storage

MTP storage provides functionality with which users browse and manage files on a device and transfer files to a device from a computer. MTP storage does not support all of the file formats, properties, and commands of the MTP 1.0 Specification. Instead, MTP storage implements a subset of the MTP specification that is necessary to meet the requirements for Windows logo certification. For more information about the requirements that devices must meet to qualify for a Windows logo, create an account on the [Windows Quality Online Services](http://go.microsoft.com/fwlink/?LinkID=179356) (<http://go.microsoft.com/fwlink/?LinkID=179356>) website, and see [Prepared Requirements Reports](http://go.microsoft.com/fwlink/?LinkID=179359) (<http://go.microsoft.com/fwlink/?LinkID=179359>).

Media MDB Storage

The Media MDB storage provides support for media file synchronization and qualifies devices for Windows logo certification in the portable media player (PMP) and other portable devices (OPD) categories.

Media MDB Operations and Objects

MTP command functionality in MTP Responder for Windows Embedded Compact 7 is confined to operations and object types that are necessary to support file browsing and management on the device, file transfers between a device and a computer, and synchronizing files from a computer to a device.

Media MDB MTP Operations

The following table lists the operations supported by MTP Responder by Media MDB storage.

Supported MTP Operations for the Media MDB Option

MTP operation	Description
OpenSession	Opens a new MTP session for communication between the computer and the device.
CloseSession	Closes an active session.
GetStorageIDs	Retrieves the storage ID for the storage area on the device.
GetStorageInfo	Retrieves a StorageInfo data set that describes a storage area on the device.
GetNumObjects	Retrieves a count of the number of MTP objects that are stored on the device.
GetObjectHandles	Retrieves an array of object handles that the computer can use to access the MTP objects that are stored on the device.
GetObjectInfo	Retrieves the ObjectInfo data set for the specified MTP object on the device.
GetObject	Retrieves the binary data component of the specified object on the device.
DeleteObject	Deletes an object or set of objects from the device.
SendObjectInfo	Sends an ObjectInfo data set to the device to prepare it to receive a new object.
SendObject	Sends the binary data component of an MTP object to the device.
FormatStore	Provides a method to quickly delete all content on the device. Using this command is faster than individually deleting each object in the

MTP operation	Description
	store.
MoveObject	Changes the location of an object on the device by changing the location in which it is located.
WMPReportAddedDeletedItems	Creates a list of objects that have been moved.
WMPReportAcquiredItems	Creates a list of objects that have been acquired.
GetDeviceInfo	Retrieves a DeviceInfo data set that describes the device in detail.
GetDevicePropDesc	Retrieves a DevicePropDesc data set for a property of the device.
GetDevicePropValue	Retrieves the current value of a property of the device.
SetDevicePropValue	Sets the current value of a property of the device.
ResetDevicePropValue	Resets the current value of a property of the device.
GetServiceIDs	Retrieves a list of the IDs of the services available on the device.
GetServiceInfo	Retrieves the ServiceInfo Data set for a particular service.
GetServiceCapabilities	Retrieves a ServiceCapabilityList data set that contains all object format and method format information for a service.
GetServiceProperties	Gets the ServicePropertyDesc data set for the service.
GetServicePropertyList	Retrieves a ServicePropList data set that contains a list of property values from the service.
SetServicePropertyList	Sets a property of the service by using the ServicePropList.
DeleteServicePropList	Removes the properties specified in the DeleteServicePropList data set from the specified service. If a property is not removable,

MTP operation	Description
	it is returned to its default value.
UpdateObjectPropertyList	Sets the property list for a particular object that will be updated with a new binary object. Use this command to replace the binary data of an existing object.
DeleteObjectPropList	Removes the properties specified in the DeleteObjectPropList data set from the specified object or objects. If a property is not removable, it is returned to its default value.
GetObjectPropsSupported	Retrieves a list of object properties that the device supports for a class of MTP objects that share a particular object format.
GetObjectPropDesc	Retrieves an ObjectPropDesc data set that describes a particular property of a class of MTP objects that share a particular object format.
GetObjectPropValue	Retrieves the current value of an object property.
SetObjectPropValue	Sets the current value of an object property.
GetObjectPropList	Retrieves an ObjectPropList data set that contains a list of property values of the object.
SetObjectPropList	Sets the current value of an object property using the ObjectPropList data set.
SendObjectPropList	Sends an ObjectPropList data set that contains a list of property values of the object.
GetObjectReferences	Retrieves an array of object handles referenced by the object.
SetObjectReferences	Sets an array of object handles referenced by the object.

For more information about MTP operations, see the [MTP 1.0 Specification](http://go.microsoft.com/fwlink/?LinkID=183533) (http://go.microsoft.com/fwlink/?LinkID=183533).

Media MDB Object Formats

The following table lists the object formats supported by the MTP Responder Media MDB.

Media MDB Object Formats

Object format	MTP datacode	Type	Description
Undefined Object	0x3000		In this format a portable device can receive an object that is encoded in an arbitrary format, regardless of whether the device explicitly supports this format.
Association	0x3001		This format is required for synchronization with Windows Media Player.
JPEG, EXIF, JPE, JPG	0x3801	Image	Joint Photographic Experts Group image file. This image file format is used by digital cameras and portable devices that support digital photography.
GIF	0x3807	Image	Graphics Interchange Format image file.
BMP	0x3804	Image	Bit map image file.
TIF, TIFF	0x380D	Image	Tagged Image File format image file.
PNG	0x380B	Image	Portable Network Graphics image file.
WAV	0x3008	Audio	Waveform audio file format.
MP3	0x3009	Audio	MPEG-1 Audio Layer 3 format.
WMA	0xB901	Audio	Windows Media Audio format.
MPEG, MPG	0x300B	Video	Motion Picture Experts Group video file.

Object format	MTP datacode	Type	Description
MP4	0xB982	Video	MPEG-4 video file.
AVI	0x300A	Video	Audio Video Interleaved file.
ASF	0x300C	Video	Advanced Systems Format, a video container format for streaming media.
WMV	0xB981	Video	Windows Media Video format.
AAC	0xB903	Audio	Advanced Audio Coding format.
Abstract Audio Album	0xBA03		Used to group audio tracks and provide album art to the device.
Abstract Audio Video Playlist	0xBA05		Required to receive playlists from Windows Media Player.
Undefined Firmware	0xB802		Enables the device vendor to install firmware upgrades that are encoded in a proprietary data format.

You can add additional object formats that the desktop version of Windows Media Player supports, such as MIDI (.mid or .midi), by updating DeviceSettings.xml and by modifying the provided MTP storage source code. You will need to add the decoders on the device side to support playback of these formats.

Media MDB Object Properties

The following table lists the object properties that Media MDB supports for all object formats.

Media MDB Object Properties for All Object Formats

Object property	MTP datacode	Description
StorageID	0xDC01	The storage area on the device where the object is stored. In MTP storage, you designate only one StorageID on the device

Object property	MTP datacode	Description
		for the folders shared for MTP access. All MTP objects on the device have the same StorageID.
Object Format	0xDC02	An object's format determines the properties that the object will support. The format is either 0x3000 (file) or 0x3001 (folder), and is stored implicitly in the MTP file system on the device.
Protection Status	0xDC03	The status is either 0x0000 (No Protection) or 0x8001 (Read-only data); the mapping depends on whether the object is marked read-only in the MTP file system.
Object Size	0xDC04	The size of the data component of the object, in bytes. This property is ignored for folders.
Object File Name	0xDC07	The file name of the object, including the extension, but not the path.
Parent Object	0xDC0B	The object handle of the parent folder, which must be an object of the Association (0x3001) format. If the object is at the root, this value is 0xffffffff.
Persistent Unique	0xDC41	Although called the persistent unique object identifier (PUOID) in the MTP specification, the
Object Identifier		PUOID is not persistent between sessions in the MTP storage implementation. The PUOID is a 128-bit version of the object handle. It is not used by the computer. It cannot be used to manage data

Object property	MTP datacode	Description
		synchronization on the computer.
Name	0xDC44	The value returned for a file is the file name without path or extension. For a folder, the value returned is the folder name without the path. The value that is sent by Windows when creating an object is ignored.
Non-Consumable	0xDC4F	Indicates whether the object was transferred to the portable media player for storage only and is not available to be consumed (for example, played) by the portable device. The device returns 1 for files and 0 (zero) for folders. Windows does not send values for this property, but any such values would be ignored if sent.
Date Created	0xDC08	The date and time when the object was created. This value maps to the value in the MTP Responder file system on the device.
Date Modified	0xDC09	The date and time when the object was last altered. This value maps to the value in the file system.
Hidden	0xDC0D	Identifies whether an object is displayed to users or is hidden and only used by applications. This value maps to the value in the MTP Responder file system.
Association Type	0xDC05	Describes the kind of collection. A value of 1 indicates that the Association Type is a generic folder, which is the only

Object property	MTP datacode	Description
		Association Type that MTP storage supports. This value is never sent by Windows. MTP storage ignores any values sent, returns 1 for folder, and returns 0 (zero) for a file.
AssociationDesc	0xDC06	Provides additional information about the Association Type. MTP storage ignores any values sent, returns 0 (zero) for either a file or a folder. This value is never sent by Windows.

In addition, the Media MDB supports more object properties depending on whether the object format translates to an image file, an audio file, or a video file. The following table lists the object properties that the Media MDB supports for image, audio, and video object formats.

Media MDB Object Properties by Content Format

Object property	MTP datacode	Image	Audio	Video
Artist	0xDC46		X	X
Description	0xDC48			X
Representative Sample Format	0xDC81	X		
Representative Sample Size	0xDC82	X		
Representative Sample Height	0xDC83	X		
Representative Sample Width	0xDC84	X		
Representative Sample Data	0xDC86	X		
Width	0xDC87	X		X
Height	0xDC88	X		X

Object property	MTP datacode	Image	Audio	Video
Duration	0xDC89		X	X
User Rating	0xDC8A		X	X
Track	0xDC8B		X	X
Genre	0xDC8C		X	X
Use Count	0xDC91		X	X
Album Name	0xDC9A		X	X
Album Artist	0xDC9B		X	X
Bit Rate Type	0xDE92		X	X
Sample Rate	0xDE93		X	X
Number of Channels	0xDE94		X	X
Scan Type	0xDE97			X
Audio Wave Codec	0xDE99		X	X
Audio Bit Rate	0xDE9A		X	X
Video FourCC Codec	0xDE9B			X
Video Bit Rate	0xDE9C			X
Frames Per Thousand Seconds	0xDE9E			X
Key Frame Distance	0xDE9E			X
Encoding Profile	0xDEA1			X
Encoding Quality	0xDEA0			X
Audio Bit Depth	0xDE95		X	X
DRM Protection	0xDC9D		X	X

For more information about MTP object formats and properties, see the [MTP 1.0 Specification](http://go.microsoft.com/fwlink/?LinkID=183533) (<http://go.microsoft.com/fwlink/?LinkID=183533>).

Media MDB Implementation

The Media MDB storage option consists of the Windows Embedded Compact 7 Media Library and an MTP Media Library plug-in that integrates the MTP responder stack with the Media Library. For more

information about the Media Library, see [Media Library](http://go.microsoft.com/fwlink/?LinkID=228737) (<http://go.microsoft.com/fwlink/?LinkID=228737>).

MTP Media Library Plug-in

The MTP Media Library plug-in is a type of “glue layer” that acts as an interface between the MTP responder stack and the Media Library. The MTP Media Library plug-in provides the following functionality:

- Translates MTP initiator object handles to and from the object handle format of the Media Library (see the following sections for additional information).
- Converts playlists from the MTP initiator to work with the paths of media files located on the device.
- Converts representative sample formats used by MTP initiator to thumbnail file formats used by the Media Library, and vice versa.
- Synchronizes media files to device storage and tracks media file creation, deletion, and modification on behalf of the MTP initiator and the Media Library.
- Broadcasts events from the Media Library to the MTP responder stack, such as the creation of a playlist, or the addition of a new photo image.

The MTP Media Library plug-in converts incoming 32-bit MTP object handles into an internal 64-bit format that the Media Library uses to manage media files; object handle conversion functionality maintains this association in a table. The Media Library plug-in also converts the internal 64-bit Media Library object format into outgoing 32-bit MTP object handles. MTP Responder effectively uses this table to translate object handles between the format that the MTP initiator uses and the format that the Media Library uses.

For information about the source code location for the MTP Media Library plug-in, see [Specifying the Maximum Number of Entries](#). For more about the Media Library, see [Media Library](#) (<http://go.microsoft.com/fwlink/?LinkID=228737>).

Media MDB Optimization

You can modify Media MDB settings to optimize them for your device; for example, to manage memory usage and to optimize performance. To adjust Media MDB optimization settings, modify the Mediaapps.reg file, which is located at

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

The optimization setting for the Media MDB is located in the following registry key:

HKEY_LOCAL_MACHINE\Software\Microsoft\MTP\Responder\MediaMDB

This setting is described in the following table. When you build your OS design, this setting is included in the boot image to configure Media MDB memory usage.

Media MDB Optimization Settings

Name	Type	Default value	Description
MaxNumSessionEntries	REG_DWORD	10000	You specify the maximum number of entries expected. This will depend on the storage capacity of your device, the scenarios that your device supports, or other factors.

Specifying the Maximum Number of Entries

The value you specify for MaxNumSessionEntries defines the maximum number of objects that are enumerable in an MTP session. You must set this value to support the actual capability of your device.

Source Code Modification

You can modify or extend the provided MTP storage implementation to add storage features that are specific to your device. To make customized modifications and extensions to the Media MDB storage, see the MTP storage source code at the following location:

%_WINCEROOT%\Public\Mediaapps\Oak\MTP\Wince\Src\MTPObjStore

Your code modifications may extend to files in other directories, depending on your objectives and the extent of the modification you want to make. Note that Microsoft does not provide source code for the Media Library catalog item.

Programming References

Several resources are available to help you to understand the MTP Responder implementation and to guide you in making modifications to the MTP Responder source code.

For more information about the MTP Responder reference implementation, see the [Media Transfer Protocol Porting Kit](http://go.microsoft.com/fwlink/?LinkID=183529) (http://go.microsoft.com/fwlink/?LinkID=183529).

For more information about MTP operations, events, properties, and formats, see the [MTP 1.0 Specification](http://go.microsoft.com/fwlink/?LinkID=183533) (http://go.microsoft.com/fwlink/?LinkID=183533).

For more information about MTP device services extensions, see the [MTP Device Services Extension Specification](http://go.microsoft.com/fwlink/?LinkID=178887) (http://go.microsoft.com/fwlink/?LinkID=178887).

For more information about the Device Enabling Kit, see the [Windows 7 Portable Device Enabling Kit for MTP, Version 7R2](http://go.microsoft.com/fwlink/?LinkID=178864) (<http://go.microsoft.com/fwlink/?LinkID=178864>).

Conclusion

MTP Responder is based on the MTP responder in the Windows 7 Portable Device Enabling Kit for MTP, version 7R2. MTP Responder consists of an MTP responder stack component and an MTP storage component. The MTP storage component provides Media MDB support for media file synchronization and persistent metadata storage.

You can modify and extend the MTP responder stack to support additional MTP commands and properties. An MTP extension can define new MTP operations, properties, and object formats that are not part of the MTP specification. The Media MDB uses a plug-in as an interface between MTP Responder and the Media Library, and uses the Media Library to store object formats and object properties that are required for MTP. You can balance performance and memory by modifying the optimization settings of hash tables used by these components.

Additional Resources

To learn more about MTP Responder, see the companion article:

- [MTP Responder Deployment Guide](http://go.microsoft.com/fwlink/?LinkID=210234) (<http://go.microsoft.com/fwlink/?LinkID=210234>)

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

- [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 resources:

- [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.