

# Introduction to Media Transfer Protocol (MTP)

December 10, 2010

## Abstract

This paper provides information about Media Transfer Protocol (MTP) for Windows® operating systems. It provides an overview for portable-device manufacturers that want to understand the benefit of supporting MTP in their hardware. And, it outlines how MTP results in significant performance improvements over the alternative mass-storage solutions.

This information applies to the following operating systems:

Windows 7

Windows Vista®

References and resources discussed here are listed at the end of this paper.

The current version of this paper is maintained on the web at:

<http://www.microsoft.com/whdc/device/wpd/mtp-intro.mspx>

**Disclaimer:** 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. Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

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.

© 2010 Microsoft Corporation. All rights reserved.

**Document History**

Date	Change
December 10, 2010	First publication

**Contents**

Media Transfer Protocol (MTP) .....	4
A Brief History.....	4
MTP's Predecessor: Picture Transfer Protocol (PTP).....	4
PTP Limitations.....	4
Early MTP Releases.....	4
MTP Today.....	4
An Overview of MTP.....	5
Supported Device Scenarios.....	5
Special Note for the Device Manufacturer.....	6
Architecture Overview.....	6
Conceptual Model.....	8
Device Representation.....	9
Comparing MTP and Mass Storage.....	10
Basic Connectivity Advantages.....	11
Media Advantages.....	11
Synchronization Advantages.....	11
End-user Experience Advantages.....	12
Extensibility Advantages.....	12
Resources .....	12

## Media Transfer Protocol (MTP)

---

What is MTP? Why was it developed? Why should I consider using it? This paper attempts to answer these three questions and related topics.

If your company creates portable devices or writes software for these devices, this paper is written for you.

### A Brief History

#### MTP's Predecessor: Picture Transfer Protocol (PTP)

The digital still camera was one of the first portable devices adopted en masse by the consumer. One of the fundamental requirements of the camera was the need to transfer images to a PC. Early on, device manufacturers created their own proprietary protocols to handle content transfer. To do away with the need for proprietary protocols, the Picture Transfer Protocol (PTP) was proposed and adopted.

PTP is a platform- and transport-independent protocol that supports common operations like connecting to a camera and transferring content. PTP was standardized in 2000 by the International Standards Organization (ISO).

#### PTP Limitations

Around the same time that digital cameras appeared, several other portable devices gained popularity with the consumer: digital video cameras, portable media players, and cell phones (which often combined the functionality found on these other devices). As these devices grew in complexity, the limitations of PTP became apparent. For example, because PTP was designed for digital still cameras, the concept of a video, a contact, or a media file was not supported.

#### Early MTP Releases

To address the shortcomings in PTP, Microsoft® proposed the Media Transfer Protocol (MTP). MTP treats logical entities on the device (such as images, videos, and contacts) as objects. Each object has its own properties. For example, an image object can be described by its file name, size, and format properties. This simplified the abstraction of the object when manufacturers designed and tested their device firmware.

In addition, it simplified the abstraction of the device for the Windows programmer who wrote applications that interacted with these devices. Using the Windows Portable Devices (WPD) application programming interface (API), an application programmer could write one program that worked with any digital camera or portable phone that supported MTP. (Because Microsoft supplies a class driver for MTP, the application programmer can ignore low-level transport issues.)

Finally, in addition to representing logical entities on the device, MTP objects could also represent device functions and information. For example, an MTP object could represent the controls on a camera (play/record/pause) or the Short Message Service (SMS) capabilities on a cellular phone.

## MTP Today

With the introduction of Windows 7, Microsoft also introduced the concept of device services for MTP. These services let manufacturers extend their device functionality and these services provide a new set of APIs (as part of Windows Portable Devices) that let programmers discover and access the extended functionality.

An example service is the status service that reports device status including:

- Signal strength
- Battery level
- Charging state
- Storage space

Using some of the data reported by this service, device manufacturers can create a remote troubleshooting application that retrieves important device-state information for their technicians. (Note that this application would support any of the manufacturer's devices that implemented the status service in their firmware. This means that one application could access multiple device types without rewrite or revision.)

For a complete list of device services, see "MTP Device Services for Windows" in the Resources section at the end of this paper.

## An Overview of MTP

### Supported Device Scenarios

This section outlines four key features of MTP: content transfer, browsing with Windows Explorer, device control, and enabling vertical solutions.

### Content Transfer Among Devices

MTP supplies the necessary infrastructure to standardize data transfers between applications and portable devices connected to a Windows PC. It provides applications with a uniform view of devices and their content, as well as standardized mechanisms to access and transfer data.

For example, an application can sync content between the portable device and the PC.

Another example is the automatic transcoding to different video formats that Windows 7 supports. Both Windows Media Player sync and shell drag-and-drop to the device, take advantage of the native Media Foundation support for transcoding in Windows 7. When the user drags and drops to a device, the necessary conversion happens automatically. Both the Windows shell and Windows Media Player use MTP properties to identify supported target formats.

*Transcoding* is the conversion of a digital media file from one format to another. It requires the following:

- Constant bit rate (CBR) encoding, where the target bit rate is known in advance.
- A single audio and video stream.
- One source file and one destination file.

Transcoding does not support the following:

- Variable bit rate (VBR) or multi-pass encoding.
- Multiple audio streams or multiple video streams.
- Digital rights management (DRM)-protected content other than Advanced Systems Format (ASF) files protected with Windows Media DRM (WMDRM).
- Live streaming, such as live-to-file streaming or live-to-live streaming.

For more information about Media Foundation's Transcode API, see "Transcode API" in the Resources section at the end of this paper.

Identifying the target file format for video containers can be tricky. It is important for the responder to properly identify supported formats.

### **Browsing Device Contents Using Windows Explorer**

Through the WPD namespace, users will be provided with a rich way to apply Windows file management approaches to portable devices of any type.

### **Device Control**

As well as providing standard access to device content, MTP offers standardized device control mechanisms. This allows applications to control various device behaviors, as well as issue device commands.

For example, PC applications for mobile phones may want to send/receive messages while the phone is connected to the PC.

### **Enabling Vertical Solutions**

The MTP infrastructure offers extremely extensible device representation and control mechanisms. Vertical solution providers can use the WPD API to create enhanced user experiences that are outside of the WPD standardized set.

Examples of this include the vendor provided software suites that ship with the device.

Note that both MTP 1.0 and PTP 1.0 were fairly restricted when it came to vertical solutions, due primarily to limited datacode space for vendor extensions. However, MTP 2.0 and MTP 1.0 with device services resolve these issues.

### **Special Note for the Device Manufacturer**

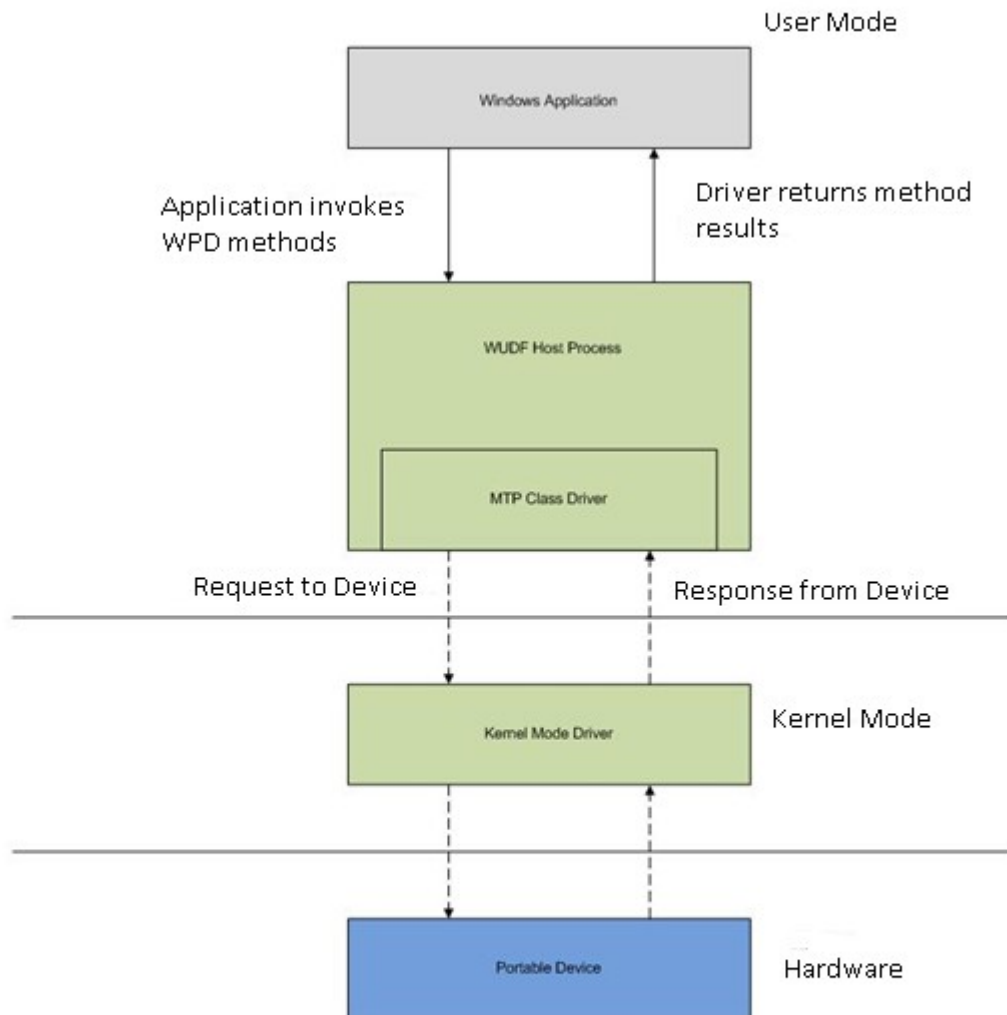
If you manufacture a portable device and require connectivity with Windows, you have the following options:

- For portable media players, digital still cameras, digital video cameras, or digital picture frames, implement support for Media Transfer Protocol (MTP). This will allow your device to work with the Microsoft MTP class driver.
- For devices that require only mass storage capabilities, you can use the WPD driver (WpdFs.sys) that ships with Windows.
- For devices that require a proprietary protocol, provide a WPD driver.

## Architecture Overview

The following illustration depicts the flow of data from a Windows application to a device. The illustration depicts a WPD method being called by the application and the response being returned by the device.

The colors in the illustration identify the author of the corresponding software. The application could be written by an independent software vendor (ISV) or the device vendor. The drivers are supplied by Microsoft. The device firmware is written by the application vendor.



**The Windows Application**

The client application uses the Windows Portable Devices (WPD) API to interact with the device. This includes:

- Moving files from the device to the PC
- Moving files from the PC to the device
- Retrieving device capabilities
- Retrieving device status
- Controlling the device

The PC running the application is referred to as the “initiator” in the documentation found in the Device Enabling Kit. (Note that the initiator could also be another device.)

**The MTP Class Driver**

Microsoft provides the MTP class driver as part of Windows 7. This driver acts as a “middleman” between the Windows application and the kernel-mode driver. The MTP class driver supports a variety of transports: Internet Protocol (IP), Bluetooth, and Universal Serial Bus (USB).

**The Kernel-Mode Driver**

The kernel-mode driver will vary depending on the transport required by the device. Its job is to translate requests from the MTP driver into transport-specific requests that are compatible with the device.

**The Portable Device**

The portable device contains firmware that is compatible with the Media Transport Protocol. (The device is known as a “responder” in the documentation found in the Device Enabling Kit.)

**Conceptual Model****Objects**

In MTP, logical entities on devices are referred to as objects. Typically, but not always, these represent data objects on the device. Objects have properties, and are referenced by object identifiers. Examples of objects include pictures and folders on a camera, songs and play lists on a media player, contacts on a cellular phone, and so on.

Objects can also represent functional or informational parts of the device. Examples of these include player controls (play/record/pause), camera settings, Short Message Service (SMS) capabilities of a cellular phone, and so on.

**Properties**

Object properties provide a mechanism for exchanging object-describing metadata. For example an image object may include properties that describe its file name, size, format, width in pixels, and so on.

Properties have a current value, as well as attributes. MTP and WPD define a set of standard properties, which make up the API and device driver interface (DDI) definitions. Vendors are not limited to the predefined properties and are free to add their own.

### Property Attributes

Property attributes describe the access rights, valid values, and other information related to a property. For example, the property representing bit rate could be a range from 8 to 20 kilohertz (kHz) with a step value of 1 kHz.

Access rights indicate whether callers can read, write, and/or delete the property. Valid values indicate restrictions for property values. Valid values are said to be of a specific form. Valid value forms include range (property can take a value from minimum to maximum with specified step), enumeration (property value is one of those in the specified list), and none (there are no specific valid values).

### Resources

Resources are placeholders for binary data. An object can have more than one resource. For example, if the object represented an image file with an audio annotation, then the resources for this object might be:

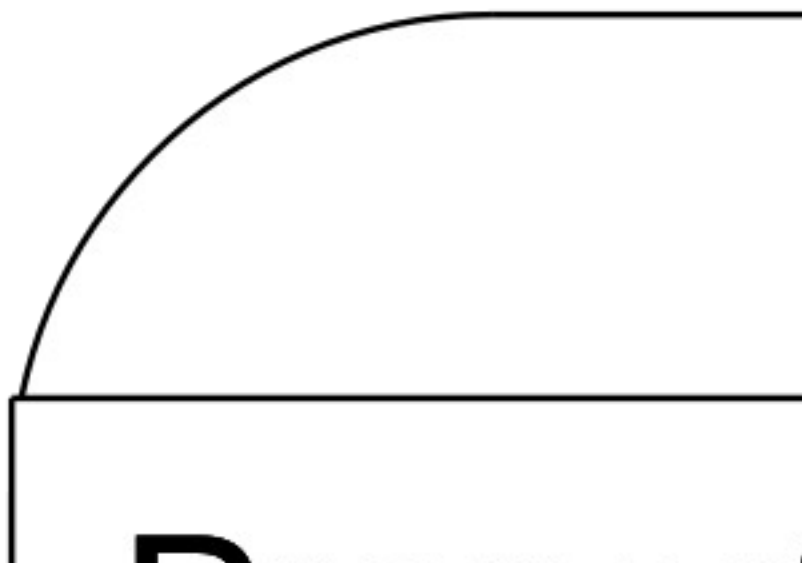
- **A default resource.** This resource represents the entire image file. (This includes any embedded data such as audio annotations, thumbnails, and so on.)
- **A thumbnail resource.** This resource represents the thumbnail data for the image.
- **An audio annotation resource.** This resource represents the audio data associated for the image.

### Resource Attributes

Similar to property attributes, resource attributes describe the access rights, size, format, and other information related to a resource. For example, the attributes for an audio annotation resource on an image object may specify the bit rate, channel count, and data format of the audio.

### Object Illustration

The following diagram shows the relationship between an object, its properties, and its resources, using an Image object as an example.



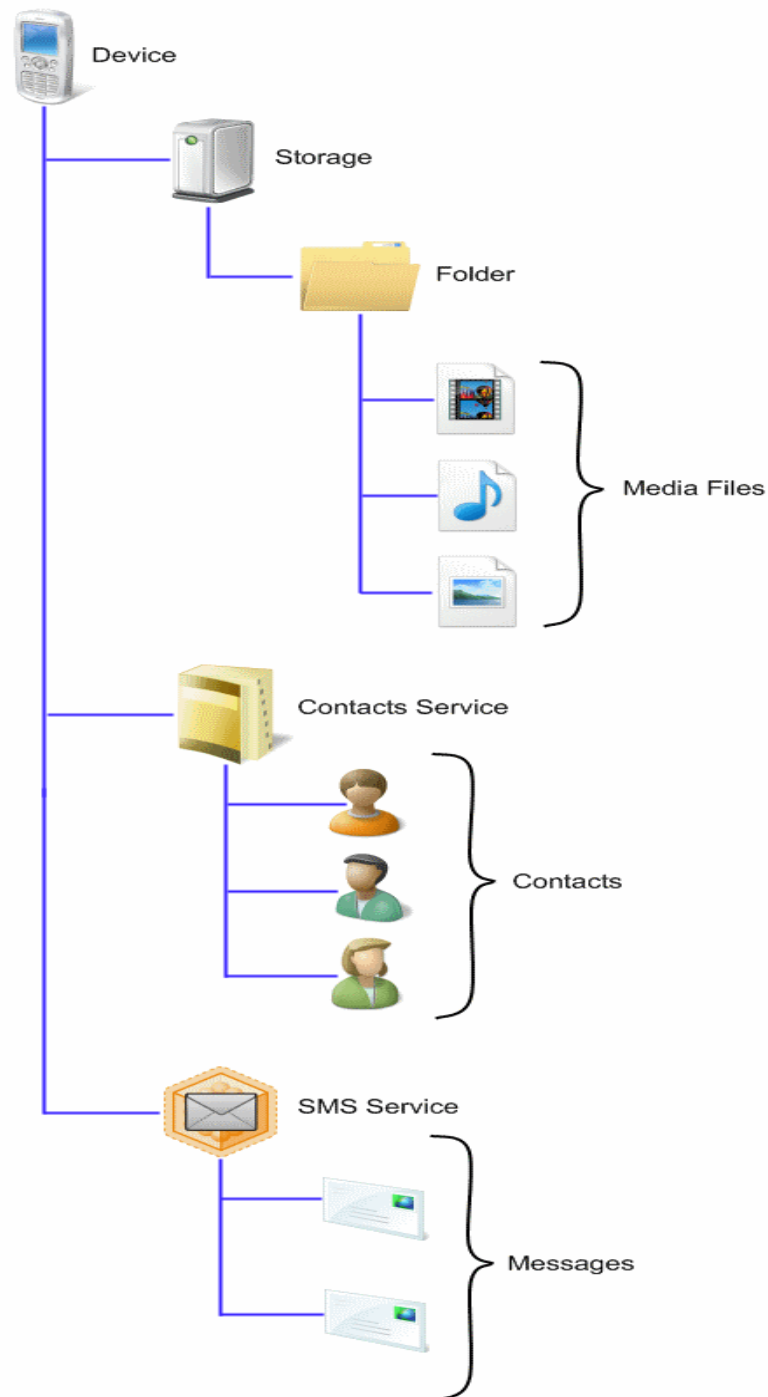


## Device Representation

Devices have two main behaviors that are addressed by the MTP architecture:

- **Accessing and storing content.** For example, applications must be able to add music files to a portable music player.
- Programming the device. This includes simple operations such as changing settings and preparing the device for data capture, or more complex operations such as uploading firmware. Examples include issuing a "take picture" command to a digital still camera.

In MTP, these behaviors are described by representing the device as a hierarchy of objects. The following figure shows an object representation for a multi-function device, which in this case is a mobile phone.



### MTP Hierarchy

This hierarchy illustrates the following functionality and contents.

Functionality:

- **Storage object.** This device has data storage.
- **Contacts Service.** This service is a functional object that can be used to synchronize and store contacts on the phone.

- **SMS Service.** This service is a functional object that can be used to send, receive, and store SMS messages.

Contents:

- **Media objects.** This device stores images, music, and video files in folders on the Storage object. Although the files shown in the illustration are stored under one folder, a device can subdivide content into folders organized by the type of media stored, for example, image folders, music folders, and video folders.
- **Contact objects.** This device stores contact information (such as name, phone number, and address) as children of the Contacts Service.
- **Message objects.** This device stores SMS messages as children of the SMS Service.

## Comparing MTP and Mass Storage

MTP provides significant advantages over mass storage solutions. These advantages fall into five categories:

- Basic connectivity
- Media support
- Device synchronization
- End-user experience
- Extensibility

These advantages are outlined in the following sections of this paper.

### Basic Connectivity Advantages

For some transports, a mass storage solution requires additional protocols. The following table compares the basic connectivity requirements of mass storage and MTP solutions.

Transport	Mass storage support?	MTP support?
USB	Yes	Yes
TCP/IP	Requires additional protocols	Yes
Bluetooth	Requires additional protocols	Yes

### Media Advantages

MTP supports common media operations that mass storage solutions do not. The following table compares support for common media operations.

Operation	Mass storage support?	MTP support?
Sync music, pictures, and video with Windows Media Player	Yes	Yes

Transcode music, pictures, and video with Windows Media Player or Windows Explorer to fit the capabilities of your device	No	Yes
Sync PlayReady and Windows Media DRM protected media to devices	No	Yes
Import pictures and videos into Windows or Windows Live Photo Gallery	Yes	Yes

### Synchronization Advantages

MTP supports device synchronization operations that mass storage solutions do not.

Operation	Mass storage support?	MTP support?
Sync Contacts, Calendar, Notes, and Tasks to a device	No	Yes

### End-user Experience Advantages

MTP supports end-user experiences that mass storage solutions do not.

Operation	Mass storage support?	MTP support?
Supports Device Stage with branding designed by the IHV	No	Yes
Display battery life and other device status	No	Yes

### Extensibility Advantages

MTP supports two extensibility features for ISVs that mass storage solutions do not.

Operation	Mass storage Support?	MTP support?
Supports access by third-party applications via the WPD API	Yes	Yes
Supports access by third-party applications via file	Yes	No

system APIs		
Supports access by websites	No	Yes
programmed with JavaScript (via Device Stage)		
Highly extensible functionality	No	Yes
Standards based	Yes	Yes

## Resources

---

### **Windows 7 Portable Device Enabling Kit**

[http://www.microsoft.com/whdc/device/wpd/MTP-DEK\\_Win7.mspix](http://www.microsoft.com/whdc/device/wpd/MTP-DEK_Win7.mspix)

### **Windows Portable Devices (WPD) API and Automation Model**

[http://msdn.microsoft.com/en-us/library/dd388998\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd388998(VS.85).aspx)

### **Windows Portable Devices (WPD) Drivers**

[http://msdn.microsoft.com/en-us/library/ff597729\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff597729(VS.85).aspx)

### **MTP Device Services for Windows**

[http://www.microsoft.com/whdc/device/wpd/mtp-devserv\\_win7.mspix](http://www.microsoft.com/whdc/device/wpd/mtp-devserv_win7.mspix)

### **Transcode API**

[http://msdn.microsoft.com/en-us/library/ff485863\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff485863(v=VS.85).aspx)