

Seamless Integration of Digital Cameras using TWAIN V1.7 and Microsoft STI Architecture

John Melville

Eastman Kodak Company, Rochester, NY USA

Abstract

Software is being developed to seamlessly integrate digital image capture devices with desktop PC operating systems. Microsoft's next operating systems, Windows 98™ and Windows NT5.0™, will provide support for still image capture devices through the Still Image Architecture (STI). STI is built on the ad-hoc industry standard for image transfer, TWAIN. The latest version of TWAIN has been extended to improve connectivity with digital cameras. The extensions to TWAIN, and the STI architecture combine to make digital cameras an integral component in this next generation of operating systems.

A brief overview of TWAIN and STI are given. The software components required to support digital cameras in these architectures are identified. The user experience with imaging devices in this framework is described.

TWAIN Background

Currently, digital capture devices, including digital cameras, are supplied with a software driver called a TWAIN data source. This software provides a set of public entry points or methods. Via these entry points a TWAIN-compliant application can select one of any installed TWAIN data sources, launch the data source, negotiate settings, provide customized user interactions with the imaging device, and transfer images.

User Interface Scenario

The user of a TWAIN imaging device launches a TWAIN-compatible imaging application. The typical imaging application has a pull-down menu from which the user executes a TWAIN "Select Source". A new dialog is displayed with a list of all the TWAIN data sources found in the system. The user selects a data source and is then returned to the imaging application.

Upon returning to the imaging application, the user initiates a TWAIN "Acquire". The selected data source is now loaded and the user is presented with the main dialog for that data source. The user interacts with the imaging device via this dialog. Settings are adjusted and one or more images are selected for transfer. Once images are selected, the user initiates transfer of the image(s) to the imaging application. The data source transfers the image(s) and then closes.

Figure 1 illustrates sample dialogs encountered in a TWAIN session and lists the components of TWAIN which produce these dialogs.

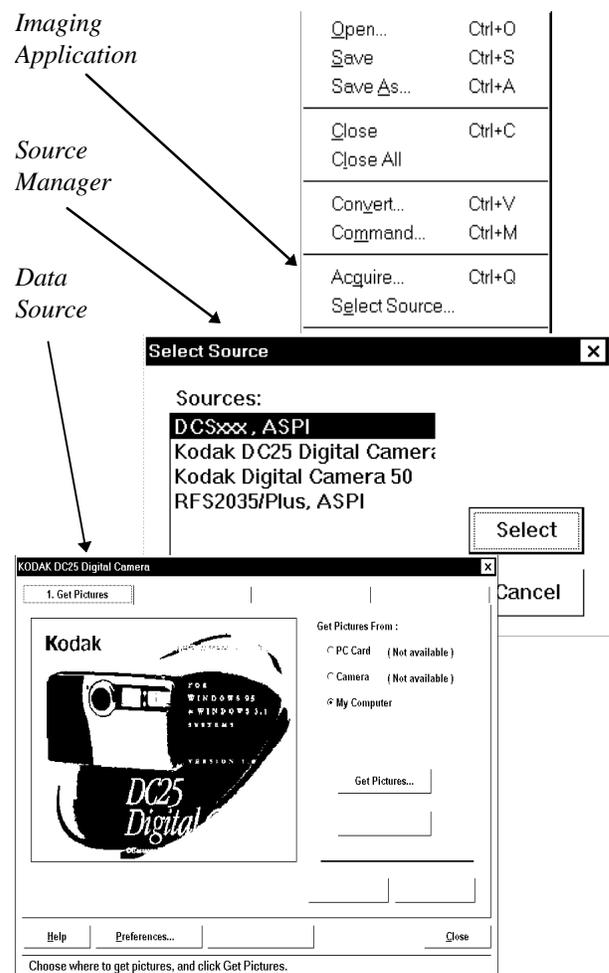


Figure 1: User Interface components of TWAIN

TWAIN Architecture

The transfer of images in TWAIN is accomplished using three software components: the application, the source manager, and the data source. The TWAIN specification describes the architecture in terms of four layers. The mapping of this architecture to the three TWAIN software components is illustrated in Figure 2.

TWAIN Application Layer

The imaging software executes in this layer. The user interface for selecting and launching a TWAIN data source

is implemented here. Apart from specification of user actions to select the data source, and the protocol followed for transfer of image data, TWAIN does not specify any other functionality of the imaging application.

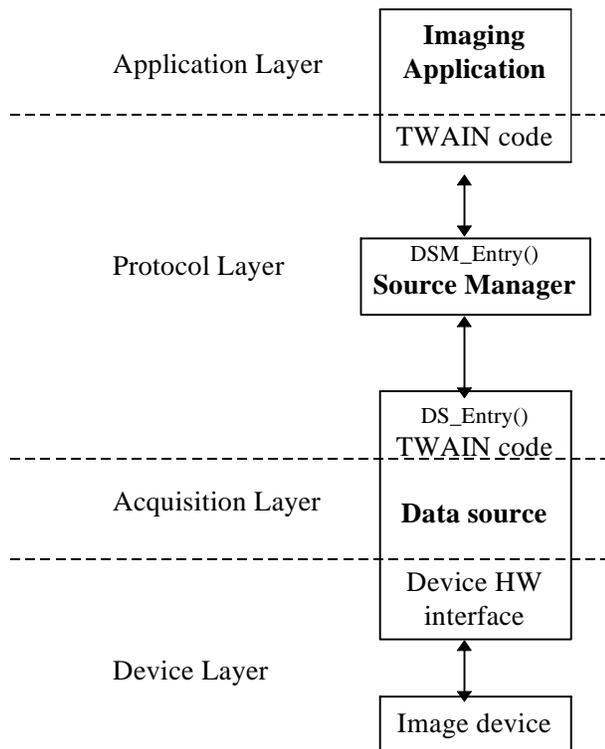


Figure 2: Software components in the TWAIN architecture

TWAIN Protocol Layer

The TWAIN protocol is comprised of a set of methods and operation triplets. It is used to discover TWAIN data source capabilities, negotiate various settings, and control the transfer of images.

TWAIN Acquisition Layer

The acquisition layer manages the imaging device and translates TWAIN protocol actions and commands to the appropriate sequence of device-specific commands.

TWAIN Device Layer

Traditional low-level device driver software is in this layer. The device driver converts the device-specific commands into hardware commands and actions which are unique to the attached imaging device. The TWAIN protocol imposes no requirements on this layer.

Communication between TWAIN Components

Communications between the three TWAIN components is accomplished by passing messages through the entry point `DSM_Entry()` in the Data Source Manager, and `DS_Entry()` in the Data Source. Return codes for success or failure are provided at the completion of each call into `DSM_Entry()` and `DS_Entry()`. Messages in TWAIN are

composed of operation triplets. The triplet fields are defined in Table 1.

In addition to the operations triplets, the TWAIN entry points also accept pointers for the message origin, message destination, and message data.

Table 1. TWAIN operation triplets

Field	Usage	Typical values
DG	Data Group	CONTROL, IMAGE
DAT	Data Argument Type	SETUPMEMXFER, IMAGEINFO, STATUS, USERINTERFACE, CAPABILITY, IMAGEMEMXFER
MSG	Message	OPENDS, GET, SET, XFERREADY, CLOSEDS

TWAIN operations are classified into two data groups (DG). CONTROL operations control the TWAIN session. Messages in this group are accepted by both the Source Manager and the Data Source. IMAGE operations indicate the kind of data to be transferred. This includes not only the image data, but also various data for setting parameters of the transfer.

The Data Argument Type (DAT) field of the operation triplet is used to identify the type of the data structure pointed to by the message data.

Messages (MSG) are used to communicate the action to be taken. Actions may be performed on the message data, or actions may be an operation which requires no data.

For the context of this discussion, it is necessary to further examine the `DAT_CAPABILITY` data argument type. `DAT_CAPABILITY` is a member of the IMAGE data group. It provides the application with the ability to GET and SET the capabilities of a data source. A `TW_CAPABILITY` data structure is pointed to by the message data pointer of this message. A key element of the `TW_CAPABILITY` message is the specification of the capability to be retrieved or set. Table 2 lists some of the capabilities which a TWAIN data source can support.

Table 2. TWAIN Example Capabilities

Capability	Usage
ICAP_BRIGHTNESS	Source brightness value
ICAP_CONTRAST	Source contrast value
ICAP_COMPRESSION	Image compression scheme
ICAP_XFERMECH	Transfer mechanism for image
ICAP_XRESOLUTION	Image x-resolution
ICAP_YRESOLUTION	Image y-resolution

TWAIN V1.5/1.6 drawbacks

TWAIN was originally devised to accommodate scanners. When used with a scanner, TWAIN provided the imaging application with the ability to acquire scanned images without presenting any additional dialog to the user. When the user launched a scanner TWAIN data source it

could automatically generate a scanned image and transfer it to the application.

This mode of operation was impossible to support for digital cameras. No mechanism existed to obtain thumbnails for images stored in the camera, and no mechanism existed to specify which images to acquire. If the data source was launched with the user interface suppressed, there was no means to indicate which image(s) to transfer. Consequently, digital camera data sources always had to provide a user interface.

In general, the data source dialog for each digital camera is unique. The requirement to present a user interface for each digital camera has detracted from the overall user experience. The result has been an inconsistent experience for the user and a more complex sequence of actions required to obtain images from the camera.

TWAIN V1.7

TWAIN V1.7 solves the problem of inconsistent dialogs between application and data source by introducing two new capabilities. Together these capabilities eliminate the principal need for a user interface in the digital camera data source.

New capabilities in TWAIN V1.7 enable the imaging application to:

- discover the number of stored images for a data source
- retrieve thumbnails for each of the stored images, and
- specify which images to transfer

Table 3 lists the key TWAIN V1.7 capabilities. For both capabilities, the data source must support MSG_SET, MSG_GET, MSG_RESET operations.

Table 3. TWAIN V1.7 capabilities

Capability	Usage
IMAGEDATASET	Gets or sets the image indices that will be delivered during standard image transfer. Indices are numbered (1..n)
THUMBNAIENABLED	Turns on/off thumbnail mode. Images transferred thereafter are sent at thumbnail size.

A TWAIN application, operating with a data source for a digital camera which supports these capabilities, can launch the data source with the source's user interface suppressed. It discovers how many images are stored in the source using the capability IMAGEDATASET with the operation triplet:

DG_IMAGE/CAPABILITY/MSG_GET

Thumbnails for each image are retrieved using the capability THUMBNAIENABLED with the operation triplet:

DG_IMAGE/CAPABILITY/MSG_SET

The application displays these thumbnails in its user interface. Still within the application, the user indicates the images to be transferred and then initiates the transfer. The application indicates the set of images to transfer using the capability IMAGEDATASET with the operation triplet:

DG_IMAGE/CAPABILITY/MSG_SET

By adding two new capabilities to the TWAIN protocol, a TWAIN application is able to select and acquire images from a digital camera TWAIN data source without ever leaving the dialog of the application.

STI

To complete the seamless integration of digital cameras, two more requirements need to be addressed. The first requirement is to provide the user a means to access and manipulate various camera settings. The second requirement is to provide a simple installation mechanism for the software supporting the camera.

Both of these requirements are satisfied by modifying the digital camera TWAIN data source to support Microsoft's Still Image Architecture (STI).

Analogy to Printers

The functionality delivered in STI can be described in terms of an analogy to the functionality delivered in Windows 95 for support of printers. In Windows 95, printers are added to the system via the "Add New Hardware" applet of the Control Panel. To install a new printer the user launches this applet and specifies "Printer" as the type of device being added. A dialog is presented with a list of manufacturers. For each manufacturer, a list of all supported printers is shown. The user selects from this list the printer to be installed.

Once installed, a control panel for the printer can be accessed from applications when preparing to print from an application, or via the "Printers" applet of the Control Panel. For any printer, the control panel has a default set of operations presented to the user. These operations can be extended to accommodate the unique attributes of individual printers.

Windows 98 and NT5.0 have added a new class of devices called "Image Devices". Like the "Printer" class, "Image Devices" are added to the system using the "Add New Hardware" applet. Once a digital camera has been installed, a "Scanners and Cameras" applet can be used to check the status of the device and to perform diagnostics. Like the "Printers" applet, the "Scanners and Cameras" applet is found in the Windows Control Panel. The default control panel behavior is extensible. Additional dialog tabs can be added to support configuration and control functions which are specialized for the installed digital camera.

STI Architecture

All of the major components of STI are shown in Figure 3. The components of interest in this discussion are the Still Image Control Panel, STI DDI, User Mode Minidriver and the TWAIN components. Note that the

TWAIN data source and TWAIN application are integral parts of this architecture.

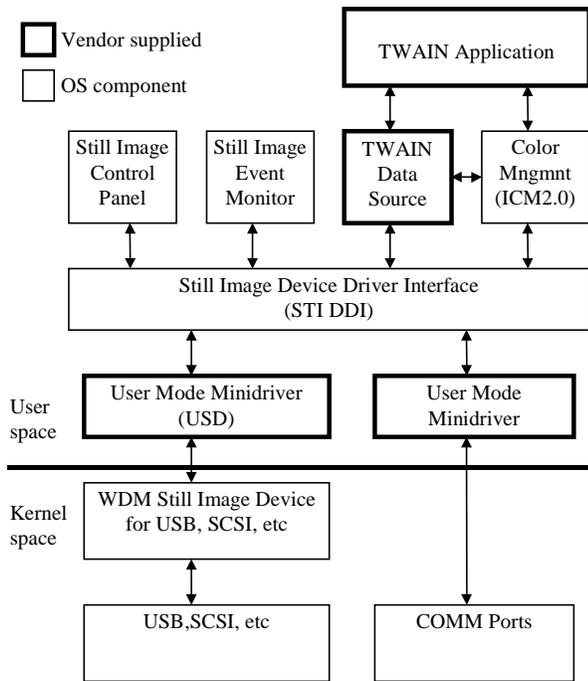


Figure 3: STI Software Architecture

TWAIN Application

As defined before, the TWAIN application is the imaging software. It is the consumer of still images.

Still Image Control Panel

Imaging devices are accessed through this control panel. The still image control panel includes the following functions:

- List of installed still image devices
- Ability to add/remove still image devices
- Test for availability of the selected device
- Vendor specific configuration and control
- Control of the association between specific device events and the application to be launched by those events.

The Still Image control panel is extended to support vendor specific configuration and control functions by using additional property pages. Adding a property page is accomplished by compiling a property page .DLL (dynamic link library). The Still Image control panel is informed of the extension through information provided during the device installation process. For each imaging device vendor, a .INF text file is included. The .INF file has a section for each supported imaging device. Entries in each of these sections contains all of the file and configuration information needed to install the imaging device. To extend the Still Image control panel for an imaging device an entry similar to the following is placed in the .INF file under the specific device for which the property page is associated.

```
PropertyPages="cameractrl.cpl,EnumStiPropPages"
```

This entry directs the Still Image control panel to load the "cameractrl.cpl" .DLL and execute the property pages extension EnumStiPropPages().

Figure 4 illustrates an example of the Still Image control panel followed by the digital camera properties dialog that would appear when the user clicks on the device properties. In Windows 98, the Still Image control panel is labeled "Scanners and Cameras Properties".

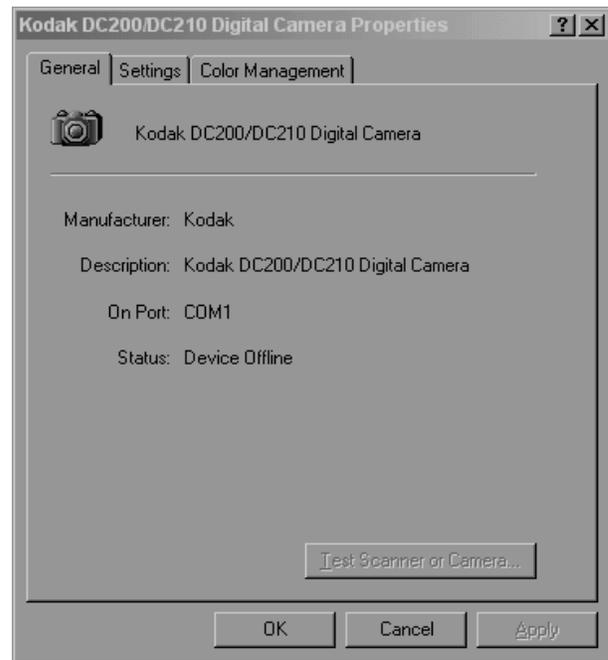
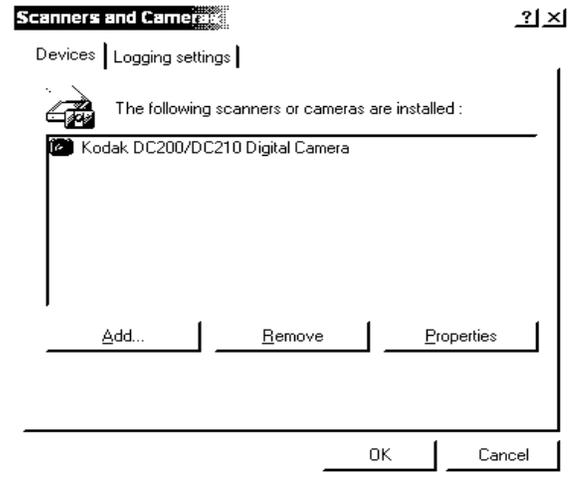


Figure 4: Still Image control panel dialogs

Still Image Event Monitor

This component is a separate application used to support push-model behavior from the imaging device. The push-model enables events which occur in the device or the hardware interface to cause the system to launch programs. The still image event monitor does the following:

- Detect events from installed still image devices
- Launch applications associated with specific events.
- Convey information to the application regarding which event was detected and which device caused the event.

TWAIN Data Source

The TWAIN data source manages the imaging device and translates TWAIN protocol actions and commands to the appropriate sequence of device-specific commands. Because the TWAIN data source is now operating in STI its bottom edge must be modified to send the device specific commands via STI DDI. Before STI, the TWAIN data source was written to access system services directly.

Color Management API

This component maintains color profiles for each imaging device and provides conversion to the desired output color space. If the installed image device has not been characterized for a color space, STI is configured to process images from the device using the default color space (sRGB).

Still Image Device Driver Interface (STI DDI)

STI DDI is an object oriented interface. Figure 5 diagrams this interface in more detail. The STI DDI consists of two interfaces: `IStillImage` and `IS StiDevice`. For each interface, several methods are defined.

`IStillImage` is used to discover what imaging devices are installed on the system, obtain details on each of the devices, and create an object for each device.

`IS StiDevice` is used to interact with a specific imaging device. Methods in `IS StiDevice` are defined to obtain device status, perform diagnostics, read or write data, and subscribe to device events.

the minidriver provides functions for test, status, and data I/O operations. These device specific functions are implemented using the standard WDM calls `CreateFile()`, `ReadFile()`, `WriteFile()`, and `DeviceIOControl()`.

Kernel WDM Still Image Device Driver

The Windows Driver Model (WDM) is a new standard for Windows device drivers. For the purposes of this paper, the WDM Still Image Device Drivers can be considered simply a way of packaging image data or commands for delivery on a specific bus.

Bus Drivers

The bus drivers perform the actual communication with the still image device. USB, SCSI, serial, and parallel are typical bus drivers.

Summary

Historically, TWAIN applications have been devoted to image editing. As applications have become more sophisticated, the use of images in PC applications has broadened. Applications such as word processors, presentation tools, chart building tools, etc. now need to import images. To achieve acceptance, these software applications must be able to import images effortlessly. The architecture to support usage of images on the desktop PC must have a minimal impact on the existing TWAIN applications but also provide a seamless interface for new applications.

TWAIN data sources for digital cameras can achieve this seamless integration in Windows 98 and Windows NT5.0 by combining the functionality of TWAIN V1.7 with the Microsoft STI architecture.

Acknowledgements

The author would like to acknowledge contributions of his associates at Microsoft and Kodak. From Kodak contributions were provided by Karl Hoeflein, Tim Looney, and Ken Parulski. From Microsoft instruction in STI, hands-on development, and consultation were provided by Robert Gross, Vladimir Sadovsky, and Dave Parsons.

References

1. TWAIN Working Group, "TWAIN, Linking Applications & Images", Release V1.5, (5/93).
2. TWAIN Working Group, "TWAIN, Linking Applications & Images", Release V1.7, (6/97).
3. Microsoft Corporation, "Memphis Beta Driver Kit, Still Image Capture Architecture and Interface", (1/97).
4. Microsoft Corporation, "WDM Still Image Architecture", (4/97)

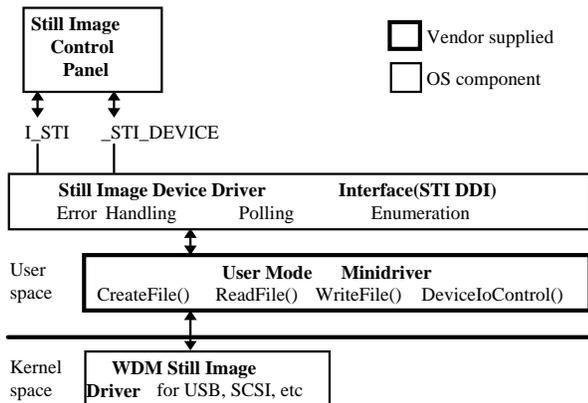


Figure 5: STI DDI details

User Mode Minidriver

This is a vendor written component. It implements the STI DDI functionality which is device specific. Specifically