**White Paper:**

# The Origin of TWAIN

The TWAIN Working Group

November, 2009

**TWAIN: Linking Applications and Images**

The TWAIN Working Group

Originally Published: March 17, 1992
Last Revised: November 19, 2009

## What TWAIN Is

TWAIN is a standard software protocol and application programming interface (API) that regulates communication between software applications and image acquisition devices such as scanners and digital cameras.

## History: The Issues that started TWAIN

In desktop publishing's early days, most publications contained only text and simple black-and-white line drawings that were output to black-and-white laser printers. As time evolved computer hardware and software became much more sophisticated enabling business professionals and graphic artists to create and output complex, full-color publications. This near-commercial-quality work included black-and-white, grayscale, and color images acquired from a wide-range of image acquisition devices. Advances in technology meant vendors were faced with a significant challenge: to supply customers with hardware seamlessly for an efficient, easy-to-use computing process. Unfortunately, image acquisition was a difficult process. It was time-consuming and tedious, and not how business professionals, designers, or publishers wanted to work. With an increasing need for on-demand integration of images acquired in real time, the image acquisition process was unacceptable.

## History: How TWAIN Provided a Solution

When the image-acquisition issue surfaced, hardware and software developers began defining their own image acquisition interfaces. This was a step in the right direction, but it soon became apparent that having a high numbers of proprietary interfaces were not the ultimate solution. It was inefficient to require a software developer to write a driver for each device they need to support. Conversely, it did not make sense to ask a hardware vendor to write a different driver to interface with each software application. Most importantly, it was not acceptable for users to have to deal with many unique application/device driver files.

Users needed a painless way to get image data into their applications. Software developers needed compatibility with the widest range of output devices without writing and maintaining

multiple device drivers. Hardware developers needed compatibility with the greatest number of applications without application-dependent coding.

The solution to this situation was an open industry interface that directly acquired image data from external sources while within an application. With this, each software developer could support a standard data acquisition manager and each hardware vendor could write one driver for their device. Hardware vendors would benefit because they needed to provide one standard driver for their device, which could then be used by all software applications supporting the standard data acquisition interface. Software vendors would be freed from writing and supporting device drivers, or from soliciting support for their own proprietary interface. Software vendors would also benefit because one single interface would support those vendors writing these device drivers. The users would also benefit because there would be a smaller set of drivers at the operating system level and there would be seamless image acquisition from a large number of applications.

## History: Forming the TWAIN Consortium

In early 1990, the formation of the Macintosh Scanner Roundtable group heightened the imaging industry's awareness of the need for an open interface. While participation in the roundtable was high, it was difficult to resolve issues and progress was slow. At one of the group's last meetings in 1990, it was suggested that a set of industry leaders form a consortium and create a specification for review, revision, and ultimate adoption by the imaging industry.

The Working Group's primary goal was promoting imaging products through developing an easy-to-use image acquisition interface and educating users about it. A key requirement of participation was that members be willing to represent a broader interest than that of their own company so that they represent a wide spectrum of application developers and hardware manufacturers. The result was that working group members then and now represent diversity in the industry bringing in-depth imaging experience to both hardware and software development.

At the TWAIN Working Group's first meetings, engineers faced the huge task of reviewing specifications and resolving outstanding requirements issues. Most Working Group companies had written their own interface for direct image acquisition, and these were considered, as well as more than two dozen specifications provided by other companies. No single specification or protocol, including those from operating system vendors, provided the completeness, richness of functionality, and ease of implementation that was required.

TWAIN Working Group engineers participated in monthly workshops to define the specification. A separate Marketing Working Group met to discuss publishing a toolkit, supporting the interface, and other issues, including how to inform the industry and public about the interface.

Besides the TWAIN Working Group, more than 175 major imaging hardware and software companies form a group called the "TWAIN Coalition." This larger group reviewed the TWAIN

specification and provided feedback prior to its release. The TWAIN Working Group took comments and suggestions from the TWAIN Coalition, examined the costs and benefits of the modifications, and decided which to incorporate into the specification. The Specification continues to be revised and updated to reflect on on-going needs of the image acquisition industry.

## Goals of the TWAIN Specification

TWAIN was designed to provide a consistent, easy integration of image data between sophisticated input devices and software applications. The following remain the Working Group's goals for the specification:

- Multiple platform support - The interface must work across many Operating Systems. Use of platform-specific mechanisms should be kept to a minimum.
- Support for multiple devices - These include digital cameras and scanners ranging from personal to high volume production.
- Widespread acceptance - Provide an interface that is well-defined and enables the majority of the leading hardware and software developers to provide drivers for their devices or include support through their applications.
- Extensibility and revisions - As the industry grows, the specification's architecture should be extensible and able to handle new, unknown conditions.
- Backward compatibility – Al revisions will be backwards-compatible with code written to earlier versions of the specification.
- Easy implementation - The interface, its documentation and sample code should be clear, well structured, well written, and intuitively designed for developers to learn and write the code for it.
- Longevity – The on-going goal of the Working Group is to provide a single solution for the industry that will last for many years.
- Multi-data Capacity - This general data interchange mechanism must be able to transmit data in existing and future file formats.
- Future technologies – the ongoing mission is to enhance the standard to accommodate future technologies.
- Public standard – TWAIN solicits feedback from the imaging industry. The source manager is open source under the LGPL license encouraging the contribution of the developer community.

## Architecture

TWAIN provides a simple methodology for universally connecting TWAIN-compliant applications with TWAIN-aware devices. The model for how applications interact with the source of input data can be described through a four-layer protocol: the Application Layer,

Protocol Layer, Acquisition Layer and Device Layer. The acquisition components correspond to these layers and include the application, Source Manager, Source and physical hardware device. The application communicates through the Source Manager to a Source driver which represents the physical hardware device that generates image data.

The hardware interface element of the TWAIN architecture is the Source. A Source is a TWAIN entity whose purpose is to get data from a hardware device and provide it to a TWAIN-compliant application. A Source is typically written by the hardware vendor to control their peripheral device. These devices are usually a physical piece of hardware, such as a scanner, although a virtual device (such as an image database) also fits the Source model. The device may be locally connected or remotely accessed over a network.

The central element of this architecture is the Source Manager (SM). The SM's primary role is to establish and manage the connections between the application and the sources. The SM allows the user to select the desired source, loads and unloads the selected source, and makes sure that all calls from a particular application are correctly routed to the appropriate source. The SM is implemented as a shared library on the Macintosh and a Dynamically Linked Library (DLL) under Microsoft Windows. When the application needs to communicate with a Source, it calls the SM with a correctly addressed message. An application never calls a Source directly.

To acquire an image using the TWAIN protocol, the user first chooses "Select Source" from the Application's menu. "Select Source" identifies the source device from which the image will be acquired, and is only accessed by the user when it's necessary to switch between connected devices. "Select Source" invokes the Source Manager and a selection dialog box appears. Once the source is selected, the user chooses "Acquire" to initiate the image acquisition process. "Acquire" brings up the Source Manager to facilitate a process called capabilities negotiation. This takes place between the application sending messages (describing the data it wants) and the Source (defining the data it can provide). When negotiation is complete, control is passed to the Source. The Source's user interface allows the user to select scanning options and start the scan or the user interface can be suppressed leaving whatever has been previously negotiated as the settings.