

## 1. Purpose

This document identifies corrections, omissions and updates that the TWAIN Working Group proposes to make in a future version of the Specification. These items related to the current 2.0 version of the TWAIN Specification

## 2. Corrections

### 2.1. DAT\_CALLBACK

TWAIN Applications running on any Linux or Mac OS X system are required to use DAT\_CALLBACK to receive asynchronous notifications of items like MSG\_XFERREADY or MSG\_CLOSEDREQ. (note that the version of DAT\_CALLBACK for Mac OS X under 1.91 varies slightly from the one specified in TWAIN 2.0, though full backwards compatibility is provided in TWAIN 2.0).

TWAIN Applications running on 64-bit Windows are strongly encouraged to use DAT\_CALLBACK, but this is not required.

TWAIN Applications running on 32-bit Windows (either natively or through WOW64) are encouraged to use DAT\_CALLBACK, but only if they call TWAINDSM.DLL. DAT\_CALLBACK is not supported by TWAIN\_32.DLL.

DAT\_CALLBACK allows asynchronous notification of DAT\_NULL events, without requiring a Windows style message loop in the Application. The TWAIN 2.0 Specification does not emphasize this enough.

## 3. Omissions

### 3.1. Open Source Licenses

The DSM is covered by the GNU Lesser General Public License (LGPL) version 2.1. The Sample Application and Sample Driver are covered by the Modified BSD License. Information on open source software licenses are available at the Free Software Foundation: <http://www.fsf.org>.

### **3.2. Linux 32-bit Support**

TWAIN 2.0 supports Linux. The Data Source Manager (DSM), Sample Application and Sample Driver have been built for various distros. RPM and Debian package files can be obtained through the TWAIN Working Group website: <http://www.twain.org>.

The DSM is called libtwaindsm.so, and is located in the /usr/local/lib directory. The TWAIN drivers (.ds files) are located in the /usr/local/lib/twain directory.

### **3.3. Linux 64-bit Support**

64-bit support for Linux is supported, but neither DSM or sample packages have not been built by the TWAIN Working Group for any existing platforms.

### **3.4. Mac OS X Support**

Full TWAIN 2.0 support is not available for Mac OS X at this time, though TWAIN 2.0 features that don't rely on the DSM can be used by both applications and drivers.

The TWAIN Working Group is investigating solutions. TWAIN 1.91 is still supported, and as of this writing (August 2008) the TWAIN DSM is still provided on all installations of Mac OS X.

### **3.5. Windows 32-bit Support**

Full TWAIN 2.0 support is not available through 16-bit TWAIN.DLL or through the 32-bit TWAIN\_32.DLL file that Microsoft delivers with its supported versions of Windows.

Full TWAIN 2.0 support is available using the new TWAINDSM.DLL provided by the TWAIN Working Group. This file is installed by a TWAIN application or driver in the SYSTEM32 directory on native 32-bit Windows systems, or in the SYSWOW64 directory on native 64-bit Windows systems.

Applications that use TWAINDSM.DLL are able to access 32-bit TWAIN drivers of any version, they are not limited to just TWAIN 2.0 drivers.

TWAIN drivers that access TWAIN\_32.DLL do not have to be modified. TWAINDSM captures accesses to DSM\_Entry through TWAIN\_32.DLL, and automatically redirects them to the DSM\_Entry in TWAINDSM.DLL.

### **3.6. Windows 64-bit Support**

TWAIN supports native 64-bit for x64 platforms. The new DSM is required, and should be installed in the Windows SYSTEM32 directory.

### **3.7. Memory File Transfers**

TWAIN supports a new CAP\_XFERMECH: TWSX\_MEMFILE, and a new operation DAT\_IMAGEMEMFILEXFER. These specific values are explained, but they are missing from the general discussion of how to transfer images from a driver to an application. The new transfer mechanism is optional, like TWSX\_FILE.

## **4. Updates**

### **4.1. Internationalization**

This section in the Specification details how to use code pages and UNICODE through the TWAIN API. In fact, it's a much better solution to consider that some items should never be localized: like the contents of TW\_IDENTITY.

Items that benefit from localization, like CAP\_AUTHOR, would do better to use UTF-8 as the one and only encoding for moving string data in and out of TWAIN.

### **4.2. Data Groups**

The TWAIN Specification mentions in more than one place that the high 8-bits in the supported groups should be treated as reserved fields.

These bits have been taken over by the new Data Flags (DF\_APP2, DF\_DS2 and DF\_DSM2).

While it seems very unlikely that any existing driver has created custom data groups, it is important to note that in this regard TWAIN 2.0 is breaking existing TWAIN 1.x behavior. The risk seems low, but if any drivers are using these bits, please contact the TWAIN Working Group.