Request for Proposal: TWAIN Classic Application Certification	Sep 13, 2019

Background

Historically TWAIN has had certification tools for Drivers. This is not sufficient since failure by an Application to adhere to the standard forces drivers to work around inconsistencies.

This is a significant gap in the TWAIN Working Group certification process and the plan is to address it with a tool that can detect and surface Application problems.

Proposal

Develop an open source Virtual TWAIN Driver (herein VDS) that has broad support for the most common scanner features. Applications can connect to this driver to exercise their use of the TWAIN interface and be informed of errors and warnings. It is recommended to use the TWAIN Sample Data Source as a starting point. A coverage report is generated that indicates how much of the TWAIN Specification was tested by the application.

Deadline for Project Completion

February 1, 2020

Description

To accomodate a reasonable Application self certification process, we will ask the tester to provide a scanning test plan that is suitable for their application. The test plan with results and logs for each test case showing no errors will be submitted to achieve certification.

Our plan is to require these tests to be run against a minimal scanner, a robust scanner and a custom scanner that the tester can choose which features they will take advantage of if they are present.

Design is expected to be modular so it can be easily scaled with new tests and requirements and be OS independent.

Only certify Apps reporting TWAIN protocol 2.2 and above.

The VDS must provide the following basic features:

- Logging of all the session activity
    - Level of TWAIN used by app
    - TWAIN State
    - Timestamps
    - Process and Thread ID’s
    - Triplets
    - Structures
    - Performance numbers (pull rate vs ppm)
- Scanner type selection
    - Predefined minimum basic scanner
    - Predefined robust scanner
    - Ability to choose from a list of available advanced features
    - Ability to set a PPM for testing
- Test all advanced features by discovery / if used it must be compliant
- Exploratory testing - follow your test plan / tool will surface problems

The VDS must respond correctly to the following TWAIN Operations:

- All mandatory TWAIN Operations
- Optional TWAIN Operations
    - The code is designed to allow adding additional operations

The VDS must respond correctly to the following TWAIN Capabilities:

- All mandatory TWAIN Capabilities and values
    - Containers and data types match spec
- Support ICAP_XFERMECH for TWSX_NATIVE, TWSX_MEMORY, and TWSX_FILE
- Support ICAP_PIXELTYPE for TWPT_RGB, TWPT_GRAY, and TWPT_BW
- Support ICAP_XRESOLUTION and ICAP_YRESOLUTION for 50, 75, 100, 150, 200, 240, 300, 400, 600, and 1200
- Support ICAP_COMPRESSION for TWCP_NONE, TWCP_GROUP4, and TWCP_JPEG
- Optional Capability values
    - The code is designed to allow adding additional enumerations to a capability
- Optional Capabilities
    - The code is designed to allow adding additional capabilities

The VDS will check the following for error conditions:

- Capabilities negotiated in the incorrect order
- All operations are made in the correct states
- All types for capabilities are correct according to the standard
- Correct responses to TWAIN events (DAT_NULL / Callbacks)
- Correct behavior to TWAIN return codes (sequence error / cancel)
- Correct transfer mode is used (ICAP_XFERMECH matches DAT_IMAGE*XFER)
- Paper jam / multi-feed / all errors
- DAT_STATUS called at the correct time (after TWRC_FAILURE, not any other time)
- Check device online behavior
- Calling an optional capability without first calling one of CAP_SUPPORTEDCAPS or MSG_QUERYSUPPORT

- Calling an optional operation without first calling CAP_SUPPORTEDDATS
- Calling custom capabilities without calling CAP_CUSTOMINTERFACEGUID
- Calling custom capabilities that do not appear in the VDS
- Correct handling of IMAGEINFO in state 6 with -1 responses for TW_IMAGEINFO fields
- Blank page removal when first page is blank
- Compressed data test (JPEG / G4 / Uncompressed)
- Cancel tests
    - Cancel occurs before first image
    - Cancel occurs mid image
    - Cancel occurs mid batch
- Memory transfer tests (if used)
    - Memory transfer last buffer contains image data
        - Random number in last buffer
        - Operator needs to type the number in
        - Unused buffer also contains a different number
    - Memory transfer last buffer contains no image data
        - Unused buffer contains a number
    - Memory transfer contains different data sizes
- TWAIN Event handling
    - Poor handling of DSEVENT/NOTDSEVENT / multiple processing
    - No pumping messages in the process in windows
    - Scanning inside MSG_XFERREADY
    - Insufficient stack (Win32 call to check - need to establish what is sufficient)
    - 2.0 using callback - threads and handling / calling in the same thread? Everything from application is calling from the main thread - need to think about this

The VDS will detect the following warning conditions:

- Use of DAT_CAPABILITY/MSG_GET in state 5 and above
- Poor performance

Deliverables

For Mac, Windows, Linux platform

- 32bit and 64bit Virtual TWAIN DS
- Installer