Request #: HUTRR48
Title: Fast IDentify Online Alliance
Spec Release: 1.12
Received: 7 Aug 2014
Requester: Kevin Lynch
Company: Synaptics, Inc.
Phone: 610-393-3437
FAX: na
email: kjlynch1@gmail.com
---
CurrentStatus: Approved
Priority: Normal
Submitted: 21 Aug 2014
Voting Starts: 5 Sep 2014
Voting Ends: 12 Sep 2014
Required Voter: Nathan Sherman, Microsoft (Chair)
Required Voter: Mark Lavelle, Logitech
Required Voter: Steve McGowan, Intel

---

Summary:
---
The Fast IDentify Online (FIDO) Alliance has defined specifications for the use of Authenticators (aka tokens) used to authenticate the users identity for secure online transactions. FIDO Authenticators may be embedded in devices or be attached to a device by a end-user. The specification of a USB HID Usage specifically for FIDO USB tokens is underway and shall be published at http://fidoalliance.org/.

This submission requests the assignment of a Usage Page ID for FIDO USB HID devices.

Background:
---
The FIDO U2F Technology Working Group has defined a USB HID Usage to provide the FIDO Protocol transport mechanism between a system and a USB HID U2F Authenticator device. This specification is currently at Working Draft status and will serve to define the initial FIDO U2F protocol tokens.

Proposal:
---
1) Assign a new Usage Page ID to Table 1 section 3 to identify USB HID devices that provide FIDO Authenticator functionality. The FIDO Alliance respectfully requests the assignment of a Usage Page ID currently in the RESERVED range, specifically Page ID = 0xF1D0.

Usage Page Title: Fast IDentity Online Alliance
Usage Page ID: 0xF1D0
Usage Description: FIDO Alliance definitions for USB attached identify authenticators

2) Add a new section, e.g. section 20 and a new table, e.g. Table 25 to describe the FIDO Usage Page:

Section 20 FIDO Alliance (0xF1D0)

The FIDO (Fast IDentify Online) Alliance page provides usage definitions for devices that include Authentication features compliant with FIDO Alliance standards. The specification will be available on the FIDO Alliance website www.FIDOAlliance.org.

Table 25 FIDO Alliance Page

| Usage ID | Usage Name | Usage Type |
|---|---|---|
| 00 | Undefined | |
| 01 | U2F Authenticator Device | CA |
| 02-1F | Reserved | |
| 20 | Input Report Data | DV |
| 21 | Output Report Data | DV |
| 22-FFFF | Reserved | |

Section 20.1 Application Usages

U2F Authenticator Device: CA - A device that provides 2nd factor authentication using the FIDO U2FHID protocol.

Input Data Report: DV - Device response data compliant with U2FHID Protocol specification.

Output Data Report: DV - Device request data compliant with U2FHID Protocol specification.

```
char ReportDescriptor[34] = {
  0x06, 0xd0, 0xf1,             // USAGE_PAGE (FIDO Alliance)
```

```
  0x09, 0x01,                    // USAGE (U2F HID Authenticator Device)
  0xa1, 0x01,                    // COLLECTION (Application)
  0x09, 0x20,                    //   USAGE (Input Report Data)
  0x15, 0x00,                    //   LOGICAL_MINIMUM (0)
  0x26, 0xff, 0x00,              //   LOGICAL_MAXIMUM (255)
  0x75, 0x08,                    //   REPORT_SIZE (8)
  0x95, 0x40,                    //   REPORT_COUNT (64)
  0x81, 0x02,                    //   INPUT (Data,Var,Abs)
  0x09, 0x21,                    //   USAGE (Output Report Data)
  0x15, 0x00,                    //   LOGICAL_MINIMUM (0)
  0x26, 0xff, 0x00,              //   LOGICAL_MAXIMUM (255)
  0x75, 0x08,                    //   REPORT_SIZE (8)
  0x95, 0x40,                    //   REPORT_COUNT (64)
  0x91, 0x02,                    //   OUTPUT (Data,Var,Abs)
  0xc0                           // END_COLLECTION
};
```

A U2FHID device implements two endpoints (except the control endpoint 0), one for IN- and one for OUT transfers. The packet size is vendor defined, but the above reference implementation assumes a full-speed device with two 64-bytes endpoints.

A transaction always consists of three stages:
1. A message is sent from the host to the device
2. The device processes the message
3. A response is sent back from the device to the host

The protocol is built on the assumption that a plurality of concurrent applications may try ad-hoc to perform transactions at any time, with each transaction being atomic, i.e. it cannot be interrupted by another application once started.

U2F HID Protocol Specification (Working Draft)

U2FHID commands

The U2FHID protocol implements the following commands. These commands are not related to U2F messages, which are encapsulated and sent using the U2FHID_MSG command.

Mandatory commands

The following list describes the minimum set of commands required by an U2FHID device. Optional- and vendor-specific commands may be implemented as described in respective sections of this document.

U2FHID_MSG

This command sends an encapsulated U2F message to the device. The semantics of the data message is defined in the U2F protocol specification.

Request

| | |
|---|---|
| CMD | U2FHID_MSG |
| BCNT | 4..n |
| DATA | n bytes |

Response at success

| | |
|---|---|
| CMD | U2FHID_MSG |
| BCNT | 2..n |
| DATA | n bytes |

U2FHID_INIT

This command requests the device to allocate a unique 32-bit channel identifier (CID) that can be used by the requesting application during its lifetime. The requesting application generates a 16 byte nonce and uses the broadcast channel ID. When the response is received, the application compares the sent nonce with the received one. After a positive match, the application stores the received channel id and uses that for subsequent transactions.

The requesting application should use the broadcast channel U2FHID_BROADCAST_CID when sending a channel allocation command.

Request

| | |
|---|---|
| CMD | U2FHID_INIT |
| BCNT | 16 |
| DATA | 16 byte nonce |

Response at success

| | |
|---|---|
| CMD | U2FHID_INIT |
| BCNT | 24 (note **) |
| DATA | 16 byte nonce |
| DATA+16 | 4 byte channel ID |
| DATA+20 | U2FHID protocol version number |
| DATA+21 | Major device version number |
| DATA+22 | Minor device version number |
| DATA+23 | Build device version number |
| DATA+24 | Capabilities, high part bits 15..8 |
| DATA+25 | Capabilities, kow part, bits 7..0 |

The protocol version identifies the protocol version implemented by the device. (**) An U2FHID host shall accept a response size that is longer than the anticipated size to allow for future extensions of the protocol, yet maintaining backwards compatibility. Future versions will maintain the response structure to this current version, but additional fields may be added.

The meaning and interpretation of the version number is vendor defined.

The following device capabilities flags are defined. Unused values are reserved for future use and must be set to zero by device vendors.

| CAPABILITY FLAGS | |
|---|---|
| CAPABILITY_WINK | Device implements the WINK function |

U2FHID_PING

Sends a transaction to the device, which immediately echoes the same data back. This command is defined to be an uniform function for debugging-, latency- and performance measurements.

| Request | |
|---|---|
| CMD | U2FHID_PING |
| BCNT | 0..n |
| DATA | n bytes |

| Response at success | |
|---|---|
| CMD | U2FHID_PING |
| BCNT | n |
| DATA | n bytes |

U2FHID_ERROR
This is command code is used in response messages only.

| Response at error | |
|---|---|
| CMD | U2FHID_ERROR |
| BCNT | 1 |
| DATA | Error code |

| The following error codes are defined | |
|---|---|
| ERR_INVALID_CMD | The command in the request is invalid |
| ERR_INVALID_PAR | The parameter(s) in the request is invalid |
| ERR_INVALID_LEN | The length field (BCNT) is invalid for the request |

| | |
|---|---|
| ERR_INVALID_SEQ | The sequence does not match expected value |
| ERR_MSG_TIMEOUT | The message has timed out |
| ERR_CHANNEL_BUSY | The device is busy for the requesting channel |

Optional commands

The following commands are defined by this specification but are optional and does not have to be implemented.

U2FHID_WINK

The wink command performs a vendor-defined action that provides some visual- or audible identification a particular U2F device. A typical implementation will do a short burst of flashes with a LED or something similar. This is useful when more than one device is attached to a computer and there is confusion which device is paired with which connection.

Request

| | |
|---|---|
| CMD | U2FHID_WINK |
| BCNT | 0 |
| DATA | n/a |

Response at success

| | |
|---|---|
| CMD | U2FHID_WINK |
| BCNT | 0 |
| DATA | n/a |

U2FHID_LOCK

The lock command places an exclusive lock for one channel to communicate with the device. As long as the lock is active, any other channel trying to send a message will fail. In order to prevent a stalling- or crashing application to lock the device indefinitely, a lock time up to 10 seconds may be set. An application requiring a longer lock has to send repeating lock commands to maintain the lock.

Request

| | |
|---|---|
| CMD | U2FHID_LOCK |
| BCNT | 1 |
| DATA | Lock time in seconds 0..10. A value of 0 immediately releases the lock |

Response at success

| | |
|---|---|
| CMD | U2FHID_LOCK |
| BCNT | 0 |

DATA n/a

Vendor specific commands

A U2FHID may implement additional vendor specific commands that are not defined in this specification, yet being U2FHID compliant. Such commands, if implemented must have a command in the range between U2FHID_VENDOR_FIRST and U2FHID_VENDOR_LAST

Response:
---
<Completed by reviewers>

Notes on Approval Procedure:
---

HID WG On Line Voting Procedures

1. Votes are on a per company basis.

2. Each Review Request shall have attached a Required Voter List that is the result of recruiting by the HID Chair and submitter of members of the USB IF. Required Voter List must include the HID Chair plus 2 companies (other than the submitter) plus any others designated by the HID Chair at the Chair's discretion. The Required Voter List ensures that a quorum is available to approve the Request.

3. Impose a 7-calendar-day posting time limit for new Review Requests. HID Chair or designate must post the RR within 7 calendar days. HID Chair or designate must work with the submitter to make sure the request is valid prior to posting. Valid review request must include all fields marked as required in the template. A new template will be adopted that requires at least the following fields: Change Text, Required Voter List, Review Period End Date and Voting End Date, Submittal Date, Submitter, Review Request Title and RR Number.

4. If a RR approval process stalls, the HID Chair may call a face-to-face meeting or conference call to decide the issue. Submitter may request that this take place.

5. Impose a minimum 15-calendar-day review period on a posted RR prior to the voting period. At HID Chair discretion, changes to the RR may require this review period to restart.

6. The Chair will accept votes via documentable means such as mail or e-mail during the 7 calendar days after the close of the review period. If a Required Voter does not vote during the period, then there is no quorum and the Chair may pursue the absent required voter and extend the voting period. The Chair may designate a substitute for the absent voter and extend the voting period if necessary.