

---

# TWAIN Direct™ Specification: TWAIN Local RESTful API

Ratified October 2<sup>nd</sup> 2017  
Revision 1.0

---



## History

Date	Version	Comment
September 15 <sup>th</sup> , 2017	1.00	First version

## Notes

Notes
<ul style="list-style-type: none"><li>• (none)</li></ul>

# Contents

[History](#)

[Notes](#)

[Contents](#)

[Glossary of Terms](#)

[References](#)

[Overview](#)

[Summary](#)

[Goals](#)

[Best Practices](#)

[Client](#)

[Scanner](#)

[HTTPS, Encryption, and Digital Certification](#)

[Commands, Errors, and Timeouts](#)

[HTTP Status](#)

[invalidJSON](#)

[Timeouts and Retries](#)

[Command Timeout](#)

[Event Timeout](#)

[Session Timeout](#)

[info Command](#)

[info](#)

[version](#)

[name](#)

[description](#)

[url](#)

[type](#)

[id](#)

[device\\_state](#)

[connection\\_state](#)

[manufacturer](#)

[model](#)  
[serial\\_number](#)  
[firmware](#)  
[uptime](#)  
[setup\\_url](#)  
[support\\_url](#)  
[update\\_url](#)  
[x-privet-token](#)  
[api](#)  
[semantic\\_state](#)

## [infoex Command](#)

### [infoex](#)

[\(all properties defined by /privet/info, plus the following\)](#)

[clouds](#)  
[clouds\[\].url](#)  
[clouds\[\].id](#)  
[clouds\[\].connection\\_state](#)  
[clouds\[\].setup\\_url](#)  
[clouds\[\].support\\_url](#)  
[clouds\[\].update\\_url](#)  
[clouds\[\].semantic\\_state](#)

## [Scanner Session](#)

### [Session States](#)

## [Common Session Command Properties](#)

[commandId](#)  
[kind](#)  
[method](#)  
[params](#)  
[params.sessionId](#)

## [Common Session Reply Properties](#)

[commandId](#)  
[kind](#)  
[method](#)  
[results](#)  
[results.characterOffset](#)

[results.code](#)  
[results.events](#)  
[results.events\[\].commandId](#)  
[results.events\[\].event](#)  
[results.events\[\].session](#)  
[results.jsonKey](#)  
[results.reason](#)  
[results.session](#)  
[results.session.doneCapturing](#)  
[results.session.imageBlocks](#)  
[results.session.imageBlocksDrained](#)  
[results.session.revision](#)  
[results.session.sessionId](#)  
[results.session.status.detected](#)  
[results.session.status.success](#)  
[results.session.state](#)  
[results.success](#)  
[results.timeRemaining](#)

## **RESTful API Commands**

[createSession](#)  
[params.locale](#)  
[waitForEvents](#)  
[params.sessionRevision](#)  
[results.events](#)  
[results.events\[n\].event](#)  
[results.events\[n\].session](#)  
[getSession](#)  
[sendTask](#)  
[params.task](#)  
[results.session.task](#)  
[startCapturing](#)  
[readImageBlockMetadata](#)  
[params.imageBlockNum](#)  
[params.withThumbnail](#)  
[results.metadata](#)  
[readImageBlock](#)

[params.imageBlockNum](#)  
[params.withMetadata](#)  
[results.metadataTwainDirect](#)  
[releaseImageBlocks](#)  
[params.imageBlockNum](#)  
[params.lastImageBlockNum](#)  
[stopCapturing](#)  
[closeSession](#)

### [Sample Sessions](#)

[Scanning a single sheet, duplex, bw1, 150 dpi, no compression](#)

---

## Glossary of Terms

This section establishes the meaning of words used within the Specification.

Word	Meaning
client	The browser or native application that discovers and communicates with a scanner.
<crLf>	HTTP messages should be fully understood by reading the RFCs defining them, and ideally should be generated and parsed with well established libraries. But to clarify the format of their construction, especially the ones for media/mixed, a blue “carriage-return linefeed” tag is placed into the examples. Please note that in the media/mixed examples the <crLf> is a separator, and the ones shown in the examples are not to be considered part of the data payload, and should not be included in the Content-Length field.
JSON	A lightweight data-interchange format.
scanner	Any physical or virtual device that captures images for a client.
user	The person in control of a client and a scanner.

---

## References

This section lists standards, guides and resources cited in this document.

Word	Meaning
Base64	RFC 1341 - Refer to 5.2 Base64 Content-Transfer-Encoding <a href="http://www.w3.org/Protocols/rfc1341/5_Content-Transfer-Encoding.html">http://www.w3.org/Protocols/rfc1341/5_Content-Transfer-Encoding.html</a>
Content-Type: multipart/mixed	RFC 2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types <a href="https://tools.ietf.org/html/rfc2046">https://tools.ietf.org/html/rfc2046</a>
Google JSON Style Guide	Google JSON Style Guide <a href="https://google-styleguide.googlecode.com/svn/trunk/jsoncstyleguide.xml">https://google-styleguide.googlecode.com/svn/trunk/jsoncstyleguide.xml</a>
Google Privet 1.0	Google Privet 1.0 (mDNS TXT records and /privet/info command) <a href="https://developers.google.com/cloud-print/docs/privet">https://developers.google.com/cloud-print/docs/privet</a>
HTTP 1.1	RFC 7230 - Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing <a href="https://tools.ietf.org/html/rfc7230">https://tools.ietf.org/html/rfc7230</a>
JavaScript Reserved Words	List of reserved words <a href="http://www.w3schools.com/js/js_reserved.asp">http://www.w3schools.com/js/js_reserved.asp</a>
JSON	ECMA-404 - The JSON Data Interchange Standard <a href="http://json.org">http://json.org</a>
Language specification	ISO 639-1:2002 Codes for the representation of names of languages -- Part 1: Alpha-2 code <a href="http://www.iso.org/iso/catalogue_detail?csnumber=22109">http://www.iso.org/iso/catalogue_detail?csnumber=22109</a>  ISO Online Browsing Platform for ISO 3166 Alpha-2 codes <a href="https://www.iso.org/obp/ui/#search">https://www.iso.org/obp/ui/#search</a>  Locale is specified as language-countrycode. So English USA is "en-us".
PDF/raster	PDF Raster Documents <a href="http://pdfrafter.org">http://pdfrafter.org</a>
TWAIN Direct Sample Code	Repository for TWAIN Direct sample code <a href="https://github.com/twain/twain-direct">https://github.com/twain/twain-direct</a>
TWAIN Direct	Website for TWAIN Direct <a href="http://twaindirect.org">http://twaindirect.org</a>
UUID	A Universally Unique IDentifier (UUID) URN Namespace <a href="http://www.ietf.org/rfc/rfc4122.txt">http://www.ietf.org/rfc/rfc4122.txt</a>  TWAIN Direct UUID strings must be represented as lowercase hexadecimal values without curly brackets. Dashes separate the hexadecimal values (the numbers represent the number of hexits in each section): 8-4-4-4-12

---

# Overview

## Summary

TWAIN Direct supports direct communication between a client and a scanner, configuring the scanner and capturing images from it. The method of establishing that communication, and the way the data is transmitted is referred to in this document as the *TWAIN Direct protocol*. The TWAIN Direct commands sent by the protocol are referred to as the *TWAIN Direct language*.

The *TWAIN Direct Specification* describes the TWAIN Direct language. Please refer to that document for more information.

The *TWAIN Direct Specification: mDNS and DNS-SD* document details how to discover TWAIN Direct scanners on a LAN without support from a DNS server.

This document describes the TWAIN Direct RESTful API protocol. It assumes the client has obtained an IPv4 or IPv6 address with a suitable port number for communication with the scanner. Under TWAIN Direct a scanner may be a physical or a virtual entity.

This scanner entity uses a HTTP server, or a Web server to process the incoming commands. Communication uses HTTPS for security.

## Goals

- Describe a state machine for the client's session with the scanner, which locks access to the scanner's physical scanning mechanism.
- Describe the RESTful API used to command and control the scanner.
- Describe the RESTful API used transfer both image and metadata from the scanner to the client.
- Describe the RESTful API used to capture asynchronous events.

---

## Best Practices

While termed “best practices” the guidelines in this section are essential for both application writers and scanner vendors. They are designed to promote the best and most efficient user experience.

Both application writers and scanner vendors use the tools and certifications recommended by the TWAIN Working Group to confirm the robustness and interoperability of their TWAIN Direct products.

### Client

- A client has the responsibility of efficiently using a scanner. It only creates a session when a user is ready to scan, and closes the session when scanning is complete, so that other clients may access the scanner.
- A client does not poll a scanner, unless instructed otherwise by this TWAIN Direct specification.
- Whenever possible a client uses concurrency to improve network efficiency, this is done when transferring images.

### Scanner

- A scanner promptly responds to a client within a few seconds, at most.
- Timeouts expire idle commands or sessions.
- Concurrency from a single client is supported, so a scanner must be capable of queuing and responding to multiple commands, even if it is unable to process them simultaneously.

---

## HTTPS, Encryption, and Digital Certification

TWAIN Local achieves security through four mechanisms:

- HTTPS to encrypt data and commands in motion
- X-Privet-Token header to prevent man-in-the-middle attacks
- Image encryption to protect data at rest (optional)
- Digital signing to detect data tampering (optional, if image encryption isn't supported)

The goal is to maximize confidence that data sent from a scanner to an application are secure and unmodified. TWAIN Direct's security model for images and metadata ceases once they are in the possession of the application collecting that data..

TWAIN Direct compliant clients and scanners must use HTTPS for secure communication in their out-of-the-box configuration. This is tested by certification. For TWAIN Local, the "https=1" entry in the mDNS TXT record indicates that the scanner is ready to receive HTTPS connections.

The X-Privet-Token is described below. Its design and intention have been documented by Google as part of the open source Privet 1.0 standard.

All scanners should support digital signing of the images they generate.

Encryption is an optional feature. Communication security is achieved through multiple layers. HTTPS by itself should not be considered sufficient to guarantee security. When images are encrypted they must also be digitally signed.

---

## Commands, Errors, and Timeouts

The TWAIN Local API is RESTful in its design. The API includes the following commands, shown in the general order most clients call them:

- info
- infoex
- createSession
- waitForEvents
- getSession
- sendTask
- startCapturing
- readImageBlockMetadata
- readImageBlock
- releaseImageBlocks
- stopCapturing
- closeSession

TWAIN Local API commands are idempotent. When handling errors or timeouts, clients may resend the exact same command without risk of unintended side effects, the state of the scanner does not change.

The `/privet/info` and `/privet/infoex` commands read data, and do not require a `commandId`.

The `/privet/twaindirect/session` commands uniquely tag each command with a `commandId` property, and a `sessionId`. The scanner recognizes when a command is repeated for a session, it must not redo the command, but must return the appropriate data. For some commands, like `sendTask`, this requires cached copy of the task response until the next command is received. The current value of the session object must be returned, so that is not cached.

All TWAIN Local API calls should respond quickly to the client (in a matter of few seconds).

### HTTP Status

The HTTP status code is not a defined part of the TWAIN Local API. Status information about a TWAIN Local command is embedded within the payload of the response, and is fully described in this specification. A TWAIN Local scanner always responds with an HTTP status code of 200 to any TWAIN Local command, whether the command itself has succeeded or not.

## invalidJSON

If there is a mistake in the construction of the JSON string, the scanner responds with an error and an indication of where the problem occurred. Processing is aborted prior to any attempt to check for actions, the results property reports “false” for success, a code of “invalidJson”, and an index where the problem was detected.

The index is in characters, not bytes, so when working with UTF-8 it’s important to properly manage multibyte characters (it’s usually easiest to convert the string to Unicode before applying the index):

This example shows an error in the createSession command:

```
POST <address>/privet/twaindirect/session HTTP/1.1<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: 122<crlf>
X-Privet-Token: bAvKr1Mlf9ba5vYN3EkBTqNV+A=:636249350840618713<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "0ac07a52-3127-4876-bebe-6ecd2351f641",,,
  "method": "createSession"
}
```

In this example the whitespace is a space (UTF8-32), and each line ends with a newline (UTF8-10). There is an extra comma, and the response points to it at offset 91:

```
HTTP/1.1 200 OK<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: 234<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "0ac07a52-3127-4876-bebe-6ecd2351f641",
  "method": "createSession",
  "results": {
    "success": false,
    "code": "invalidJson",
    "characterOffset": 91
  }
}
```

## Timeouts and Retries

TWAIN Local supports three timers, one for client commands, and two for the scanner session. Defaults are recommended. Applications and scanners should allow these to be overridden by users.

### Command Timeout

The command timeout is maintained by the client application. The recommended default is 15 seconds. If this timeout expires, the application may resend the same command. This can be repeated as many times as desired by the user.

### Event Timeout

The event timeout is maintained by the scanner. The recommended default is 30 seconds. When the event expires the scanner sends a results.code of “timeout” to the application. Other events may appear in the same payload. The application takes no action on the timeout, but since the current waitForEvents command has completed, the application issues a new waitForEvents command to continue waiting for events.

### Session Timeout

The session timeout is maintained by the scanner. The recommended default is 5 minutes. The timer resets when a /privet/twaindirect/session/\* command with the current sessionId is received by the scanner. Calls to /privet/info and /privet/infoex do not reset this timer, neither do calls to createSession, or any TWAIN Local API call with a non-matching sessionId.

When this timeout expires, an event with results.code of “critical” is sent to the application, and the scanner immediately terminates the session, returning to a noSession state.

---

## info Command

TWAIN Local's use of Google's Privet 1.0 standard requires support for the `/privet/info` command. This command can be issued at any time. It returns information about the scanner. A client must send this command once, to get the `x-privet-token`, which is required for all commands sent to `/privet/twaindirect/session`.

### info

#### version

A string. Must be "1.0".

#### name

A string. Human readable name of the device. For TWAIN Local scanners this is identical to the "ty" field in the mDNS TXT field.

#### description

A string. Device description. Should be modifiable by the user. For TWAIN Local scanners this is identical to the "note" field in the mDNS TXT field.

#### url

A string. URL of the cloud server this device is talking to. Must be an empty string, or "https://www.google.com/cloudprint", if the scanner supports cloud printing. For TWAIN Local scanners this is identical to the "url" field in the mDNS TXT field.

#### type

A list of comma separated strings. List of supported device types. Must contain "twaindirect". For TWAIN Local scanners this is identical to the "type" field in the mDNS TXT field.

#### id

A string. Device id. Empty if device is not registered. For TWAIN Local scanners this is identical to the "id" field in the mDNS TXT field.

#### device\_state

A string. State of the device. **idle** - the device is ready. **processing** - the device is busy and functionality may be limited for some time. **stopped** - the device is not working, and user intervention is required.

### connection\_state

A string. State of the connection to the server (base url). **online** - connection available. **offline** - no connection. **connecting** - performing startup steps. **not-configured** - connection has not been configured yet. For TWAIN Local scanners this is identical to the "cs" field in the mDNS TXT field.

### manufacturer

A string. Name of the device manufacturer.

### model

A string. Model of the device.

### serial\_number

A string. Unique device identifier. In this spec, this MUST be a UUID. (GCP 1.1 spec) (optional) We strongly recommend using the same serial number ID everywhere, so different clients can identify the same device. For example, printers implementing IPP may use this serial number ID in "printer-device-id" field.

### firmware

A string. Device firmware version.

### uptime

A string. Number of seconds from the device boot.

### setup\_url

A string. (optional) URL (including protocol) of the page with setup instructions.

### support\_url

A string. (optional) URL (including protocol) of the page with support, FAQ information.

### update\_url

A string. (optional) URL (including protocol) of the page with update firmware instructions.

### x-privet-token

A string. Value of the *X-Privet-Token* header that has to be passed to all APIs to prevent XSS and XSRF attacks.

### api

An array. List of supported APIs. For TWAIN Direct the only required API is "/privet/twaindirect/session".

## semantic\_state

A string. (optional) Semantic state of the device in CloudDeviceState format, as defined by Google Cloud Print.

## Command

```
GET <address>/privet/info HTTP/1.1<crLf>
X-Privet-Token: ""<crLf>
<crLf>
```

## Response

```
HTTP/1.1 200 OK<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
<crLf>
{
  "version": "1.0",
  "name": "Manufacturer's description of the scanner",
  "description": "User's description of the scanner",
  "url": "",
  "type": "twaindirect",
  "id": "",
  "device_state": "idle",
  "connection_state": "offline",
  "manufacturer": "Manufacturer's Name",
  "model": "",
  "serial_number": "",
  "firmware": "",
  "uptime": "",
  "setup_url": "",
  "support_url": "",
  "update_url": "",
  "x-privet-token": "Token returned by the scanner",
  "api": [
    "/privet/twaindirect/session"
  ],
  "semantic_state": ""
}
```

---

## infoex Command

The `/privet/info` command has two significant limitations. First, being part of the Privet 1.0 standard, it may be called by any application that recognizes a device by its advertisement of `_privet._tcp`. So calls made to it cannot be limited to scanning applications. Second, the widespread use of schemas makes it possible that these applications could react in undesired ways to new fields.

Therefore, the TWAIN Working Group has opted to define a new command `/privet/infoex`, which is a strict superset of `/privet/info`. This means that every property and range of values defined for `/privet/info` is required to be a part of `/privet/infoex`. Read and understand `/privet/info` before reading about this command.

The properties in `/privet/info` that relate to the cloud are: `url`, `id`, `connection_state`, `semantic_state`, `setup_url`, `support_url`, `update_url`. These values must reflect the settings for the default cloud, if the scanner is registered with one, and that must be “`https://www.google.com/cloudprint`”, if the scanner supports it.

At the scanner vendor’s discretion the `setup_url`, `support_url`, and `update_url` may point to the local area network address of the scanner. New properties allow for the advertisement of multiple clouds.

### infoex

(all properties defined by `/privet/info`, plus the following)

#### clouds

An array. This field must be present. If the scanner is not attached to any clouds the array is empty. At no time should an entry be included unless it defines a url. If Google Cloud Print is supported the mDNS TXT fields for `url`, `cs`, and `id` must reflect it, otherwise they show the information for the the TWAIN Cloud servers. The selection of a default TWAIN Cloud is managed by the scanner vendor.

### clouds[].url

A string. Mandatory. The URL of the cloud server this device is talking to. One of the entries must be “https://www.google.com/cloudprint”, if the scanner supports cloud printing. All other entries must point to a TWAIN Cloud server. See the document on TWAIN Cloud for more information.

### clouds[].id

A string. Device id. Empty if device is not registered with this cloud. For TWAIN Local scanners this is identical to the “id” field in the mDNS TXT field.

### clouds[].connection\_state

A string. State of the connection to the server (base url). **online** - connection available. **offline** - no connection. **connecting** - performing startup steps. **not-configured** - connection has not been configured yet. For TWAIN Local scanners this is identical to the “cs” field in the mDNS TXT field.

### clouds[].setup\_url

A string. (optional) URL (including protocol) of the page with setup instructions.

### clouds[].support\_url

A string. (optional) URL (including protocol) of the page with support, FAQ information.

### clouds[].update\_url

A string. (optional) URL (including protocol) of the page with update firmware instructions.

### clouds[].semantic\_state

A string. (optional) Semantic state of the device in CloudDeviceState format, as defined by Google Cloud Print.

## Command

```
GET <address>/privet/infoex HTTP/1.1<crLf>  
X-Privet-Token: ""<crLf>  
<crLf>
```

## Response

```
HTTP/1.1 200 OK<crLf>  
Content-Type: application/json; charset=UTF-8<crLf>  
Content-Length: 512<crLf>  
<crLf>
```

```

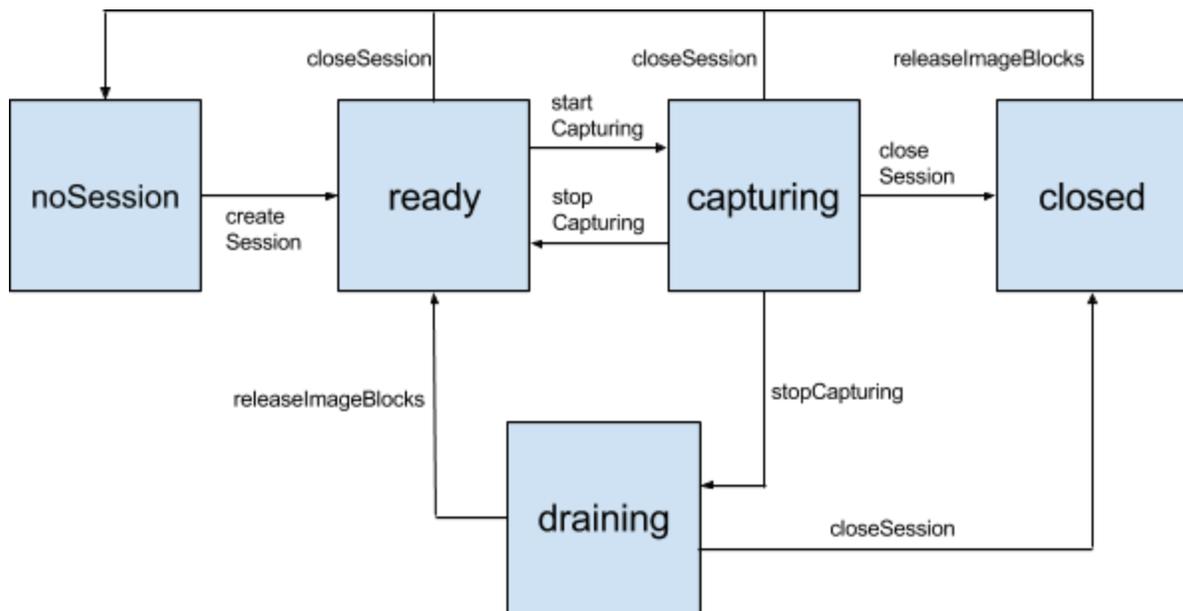
{
  "version": "1.0",
  "name": "ACME Scanner 9000",
  "description": "my scanner",
  "url": "https://www.twaincloud.com/twaindirect",
  "type": "twaindirect",
  "id": "6c12b20b-ea98-4b60-83b9-b23d074c42d1",
  "device_state": "idle",
  "connection_state": "online",
  "manufacturer": "ACME",
  "model": "Scanner 9000",
  "serial_number": "2d287b24-8a0c-44d8-bc81-e242dab42e70",
  "firmware": "1.0",
  "uptime": "27384",
  "setup_url": "",
  "support_url": "",
  "update_url": "",
  "x-privet-token": "K4xnxUphbUv/Hs9w0pJnRbyxI0o=:636245584802123447",
  "api": [
    "/privet/twaindirect/session"
  ],
  "semantic_state": "",
  "clouds": [
    {
      "url": "TWAIN Cloud URL",
      "id": "6c12b20b-ea98-4b60-83b9-b23d074c42d1",
      "connection_state": "online",
    }
  ]
}

```

## Scanner Session

A client establishes a session with a scanner to lock access to its scanning mechanism, and to continue transferring images after the scanning mechanism has been released (when this behavior is supported by the scanner).

### Session States



State	Description and Transition Rules to Get to this State
noSession	<p>The scanner is idle, and may accept a createSession command.</p> <ul style="list-style-type: none"> <li>- <i>noSession</i>: the scanner has just been initialized.</li> <li>- <i>ready</i> → <i>noSession</i>: the scanner successfully processed a closeSession command.</li> <li>- <i>capturing</i> → <i>noSession</i>: the scanner has no pending imageBlocks, and successfully processed a closeSession command.</li> <li>- <i>closed</i> → <i>noSession</i>: the last pending imageBlocks are removed by the releaseImageBlocks command.</li> <li>- <i>ready</i>, <i>capturing</i>, <i>draining</i>, or <i>closed</i> → <i>noSession</i>: the session expired due to lack of activity (sessionTimedOut event).</li> </ul>

ready	<p>The scanner is locked by a user, but is not capturing sheets of paper, or transferring images.</p> <ul style="list-style-type: none"> <li>- <i>noSession</i> → <i>ready</i>: the scanner successfully processed a createSession command.</li> <li>- <i>capturing</i> → <i>ready</i>: the scanner has no pending imageBlocks, and successfully processed a stopCapturing command.</li> <li>- <i>draining</i> → <i>ready</i>: the last imageBlocks have been removed by the releaseImageBlocks command.</li> </ul>
capturing	<p>The scanner is locked by a user, and is capturing sheets of paper, and transferring images.</p> <ul style="list-style-type: none"> <li>- <i>ready</i> → <i>capturing</i>: the scanner successfully processed a startCapturing command.</li> </ul>
draining	<p>The scanner is locked by a user, and is transferring images, but is not capturing new sheets of paper from the scanner.</p> <ul style="list-style-type: none"> <li>- <i>capturing</i> → <i>draining</i>: the scanner successfully processed a stopCapturing command, but there are pending imageBlocks.</li> </ul>
closed	<p>No more sheets of paper can be scanned as a part of this session, but there are pending imageBlocks. The scanner may or may not be locked at the discretion of the scanner vendor. The client must transfer and/or release all remaining imageBlocks.</p> <ul style="list-style-type: none"> <li>- <i>capturing</i> or <i>draining</i> → <i>closed</i>: the scanner has successfully processed a closeSession command, but there are pending imageBlocks.</li> </ul>

---

## Common Session Command Properties

This section describes properties common to all session commands sent from the client to the scanner. Their use is shown in the examples for each of the session commands.

### **commandId**

A string. A unique command id sent from the client to the scanner, to help the client pair the scanner's response with the original command. For instance using the current timestamp to at least millisecond resolution. This value is required for the application to resolve which command has finished, when a `commandComplete` event is received from the scanner.

### **kind**

A string. Indicates the type of the outermost JSON object. For command sent to `/privet/twaindirect/session`, it must be "twainlocalsession".

### **method**

A string. The command being sent to the scanner, for instance: `createSession`.

### **params**

An object. Parameters to a command. Command specific properties are described in the section for a command, for instance: `releaseImageBlocks`.

### **params.sessionId**

A string. A unique identification for the client's session with the scanner. This value is returned by the scanner as part of `createSession`, and must be used by the client in all subsequent commands during that session.

---

## Common Session Reply Properties

This section describes properties common to all session command replies sent from the scanner to the client, including errors. Their use is shown in the examples for each of the session commands and in the section on [Session Samples](#).

### commandId

A UUID in the form of a string. The commandId sent from the client to the scanner in the original command.

### kind

A string. It describes the JSON object. For TWAIN Local this must be "twainlocalscanner".

### method

A string. The method sent from the client to the scanner in the original command.

### results

An object. The results of command that the scanner is sending back to the client, with additional properties described below.

### results.characterOffset

A number, only included if results.code is set to invalidJson. This is a number from 0 - n, indicating the offset of the character that caused the JSON error.

### results.code

A string. If [results.success](#) is false a code must be included. Refer to the section on [Session Samples](#) for examples of various kinds of codes.

code	Meaning
aborted	The scanner rejected the command. For example the scanner could reject startCapturing if it requires user confirmation to proceed, and it didn't get it. The scanner must include a <b>results.reason</b> property, which is a localized string explaining how the user handles the error.
badValue	The application sent a bad value in the RESTful API (TWAIN Direct tasks report this in a different place). The scanner must include a <b>results.jsonKey</b> property, indicating the full path, in dotted notation, to the property that caused the error.
busy	The scanner is not available at this time, either another user owns it, or it is processing a command, and cannot accept another at this time.

commandPending	The command has been accepted and the scanner is working on it. This is only supported for sendTask, and only for custom vendor actions. This status is not returned by any standard TWAIN Direct command. The client must watch for an event that includes a commandId matching the original command.
critical	An internal error has occurred within the scanner. The scanner must include a <b>results.reason</b> property, which is a localized string explaining how the user handles the error.
invalidJson	The JSON data is ill-formed. The scanner must include a <b>results.characterOffset</b> property, which is a number from 0 - n, indicating the offset of the character that caused the error.
invalidSessionId	The command either did not include a sessionId, or the sessionId does not match the current scanner session.
invalidState	The command is not allowed in the current state. Use <a href="#">getSession</a> to ask for the current state of the scanner.
invalidTask	For <a href="#">sendTask</a> only. The contents of the task property are ill-formed. The scanner must include a <b>results.jsonKey</b> property, indicating the full path, in dotted notation, to the property that caused the error.
invalid_x_privet_token	The command either did not include an X-Privet-Token header, or the token in the header was invalid for this scanner.
timeout	The command expired. This should only be returned by calls to <a href="#">waitForEvents</a> . On receipt the application sends a new <a href="#">waitForEvents</a> command.
waking	The command did not succeed, the scanner is waking from a low power state. Scanner vendors are encouraged to avoid using this status. Application writers must be prepared to handle it. The scanner must include a <b>results.timeRemaining</b> property, which is a number from 0 - n, indicating the number of seconds before the scanner will be ready to process commands.

### results.events

An array. Each object in the array is an event.

### results.events[].commandId

A UUID in the form of a string. For use with vendor specific tasks that return a status of commandPending. This commandId must match the one included with the original sendTask command.

### results.events[].event

A string. The name of the event.

Status	Meaning
--------	---------

commandComplete	A sendTask command that returned a code of commandPending has completed.
commandUpdate	A sendTask command that returned a code of commandPending has updated information for the client (it is not complete).
imageBlocks	The event includes a session object. A change to the imageBlocks was the reason for the event, but the client must process all of the session fields.

### **results.events[].session**

Session information for this event. See results.session.\* for information on each field.

### **results.jsonKey**

A string, only included if results.code is set to invalidTask. The full path in dotted notation to the property that caused the error (ex: actions[0].streams[0].sources).

### **results.reason**

A string, only included if results.code is set to critical. Localized text from the scanner (default is English) describing the reason for the critical error. The application shows this string to the user, without changes and without interpretation.

### **results.session**

An object. Information about the current session between the client and the scanner.

### **results.session.doneCapturing**

A boolean. It's set to true when the scanner is no longer capturing new images. If an application is done capturing new images, it should send the closeSession command, and continue transferring images until results.session.imageBlocksDrained is set to true.

### **results.session.imageBlocks**

An array of integer numbers counting from 1 to n. Each image block represents metadata and data for part or all of an image accessible to readImageBlock and readImageBlockMetadata, and released using releaseImageBlocks. This array can be empty if the scanner is currently capturing a new image or is no longer capturing images. See results.session.imageBlocksDrained below.

### **results.session.imageBlocksDrained**

A boolean. Set to true when releaseImageBlocks frees the last imageBlock, and the scanner is no longer capturing images.

### **results.session.revision**

An integer number. A value starting at 1 that is incremented by 1 whenever the session object is updated by the scanner.

### **results.session.sessionId**

A string. A unique identification for the client's session with the scanner. This value is returned by the scanner as part of createSession, and must be used by the client in all subsequent commands during that session.

### **results.session.status.detected**

A string. The current condition of the scanner. See [results.session.status.success](#) for information on how an application should respond when detected isn't nominal. Errors associated with images are reported in the metadata, please refer to description of *metadata.status* inside of the TWAIN Direct Specifications chapter on Metadata for more info.

Status	Meaning
coverOpen	One or more doors on the scanner is not fully closed or latched.
foldedCorner	A sheet of paper that the scanner is trying to capture has a folded corner.
imageError	Catch-all condition for imaging errors not otherwise described in this table, such as low light levels from a lamp, or an uncorrectable skew in the angle of the image.
misfeed	Catch-all condition for feeder errors not otherwise described in this table, such as an inability to draw paper into the scanner.
doubleFeed	More than one sheet of paper has entered the scanner.
nominal	There are no problems with the device. If this is set, then results.session.status.success must be set to true.
paperJam	Paper is jammed in the scanner.
noMedia	The scanner cannot detect a first sheet to capture.
staple	A staple was detected on the sheet of paper, or any item that could potentially damage the scanner.

### **results.session.status.success**

A boolean. If false, the scanner requires user intervention. If true the scanner may opt to report some of these, but the application is not required to take any action. If the value of results.session.status.detected is nominal, then the value of this property must be true. See [results.session.status.detected](#) for more info.

### **results.session.state**

A string. The current state of the client's session with the scanner. See the section in this section titled [Scanner Session](#) for more information.

**results.success**

A boolean. If false, a code is included that relates to the command that was sent. Additional information is included in the [results.code](#).

**results.timeRemaining**

A number, only included if results.code is set to waking. An estimate of the number of seconds remaining until the scanner is ready to process commands.

---

## RESTful API Commands

This section describes each of the RESTful API commands supported by TWAIN Local.

Each command includes the following:

- a title
- a description
- a state transition
- the HTTP headers and command data sent from the application to the scanner
- the HTTP headers and data sent back from the scanner to the application

## createSession

Establish a new session with the scanner. On success the scanner transport is locked for use, and remains locked until closeSession is sent, or the session timeout expires.

The scanner responds immediately to this command. If it cannot respond immediately for any reason, and it's available for use, it replies with a *waking* status, and the client must try again.

If all current scanner sessions are a *closed* state, and the scanner supports multiple sessions, it can accept the createSession request. Otherwise, it must return a *busy* status.

The scanner returns a sessionId, which must be used in all subsequent command for this session.

### params.locale

A string. A locale specifier in the form *language-countrycode*. See the [References](#) section for more information. If supported by the scanner, this property overrides the scanner's default locale until closeSession is called, the session times out, or communication is lost. If for any reason the locale cannot be set, it's ignored. There is no error.

### Session State Transitions

Current State	New State	Transition Notes
capturing	capturing	Status is <i>busy</i> , remain in <i>capturing</i> .
closed	closed	Status is <i>busy</i> , remain in <i>closed</i> .
draining	draining	Status is <i>busy</i> , remain in <i>draining</i> .
noSession	ready	Transition to <i>ready</i> .
ready	ready	Status is <i>busy</i> , remain in <i>ready</i> .

### Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
X-Privet-Token: token from /privet/info command<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
```

```
"commandId": "Client generated id for this command",  
"method": "createSession"  
}
```

## Response

```
HTTP/1.1 200 OK<crif>  
Content-Type: application/json; charset=UTF-8<crif>  
Content-Length: number of bytes of UTF-8 encoded JSON data<crif>  
<crif>  
{  
  "kind": "twainlocalscanner",  
  "commandId": "commandId sent in POST command",  
  "method": "createSession",  
  "results": {  
    "success": true,  
    "session": {  
      "sessionId": "Session ID created by scanner for this session",  
      "revision": 1,  
      "state": "ready"  
    }  
  }  
}
```

## waitForEvents

Waits for asynchronous events reported by the scanner.

An application sends `waitForEvents` after a successful call to `createSession`. Unlike other TWAIN Local commands, `waitForEvents` is a long poll, and blocks until the scanner has an event to deliver, so the client must either make the call in a dedicated thread, or use callbacks.

The recommended timeout is 30 seconds. When it expires, the client immediately sends a new `waitForEvents` command. The scanner should allow the user to configure this timer.

The client includes the value of the last `sessionRevision` that it processed from any TWAIN Local command or event received from the scanner for this session. Scanners only report session objects with a `sessionRevision` greater than this value. Since other commands may be issued while waiting for events, the client only processes events containing a `sessionRevision` greater than the last one processed.

The event includes a reason for why it was triggered. The application is responsible for updating itself with all of the session object values received from the scanner.

The application must protect access to its copy of the session object. Events can be delivered at any time. An application must immediately process an event after completing work on the current TWAIN Local command. If there is no current command, it must process the event immediately.

A scanner supports one pending `waitForEvents` command per session. If a new `waitForEvents` command is received, it aborts the current one.

The scanner must maintain a queue of events. When there is a pending `waitForEvents` command all events with a `sessionRevision` greater than the one delivered by the last `waitForEvents` command are copied into the response. This prevents events from being lost if they occur in the gap between the expiration of one `waitForEvents` command and the start of a new one.

Items in the event queue are only removed when their session revision is less than or equal to the `sessionRevision` included with the `waitForEvents` command. This prevents loss of events information due to communication errors.

The [Sample Session](#) section at the end of the document contains a contrived example showing all of the eventing situations applications and scanners must handle.

### params.sessionRevision

A number. The revision value from the last session object sent from the scanner to the application. The scanner only delivers events higher than this number.

### results.events

An array. A list of one or more event objects, sorted in increasing order by the session revision number.

### results.events[n].event

A string. The name of the event that was triggered in event object number “n”.

### results.events[n].session

An object. A session object, which includes information associated with event number ‘n’. See the section above on [Common Session Reply Properties](#) for the contents of the session object.

## Session State Transitions

Current State	New State	Notes
capturing	capturing	Remain in <i>capturing</i> .
closed	closed	Remain in <i>closed</i> .
draining	draining	Remain in <i>draining</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Remain in <i>ready</i> .

## Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
X-Privet-Token: token from /privet/infoex command<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "waitForEvents",
  "params": {
    "sessionId": "sessionId from /privet/session/createSession",
    "sessionRevision": revision number of last session object processed by the application
  }
}
```

## Response

```
HTTP/1.1 200 OK<crif>
Content-Type: application/json; charset=UTF-8<crif>
Content-Length: number of bytes of UTF-8 encoded JSON data<crif>
<crif>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "createSession",
  "results": {
    "success": true,
    "events": [
      {
        "event": "imageBlocks",
        "session": {
          "sessionId": "sessionId from /privet/session/createSession",
          "revision": Current revision number for this session object,
          "state": "Current scanner state",
          "imageBlocks": [ pending image blocks (if any) ]
        }
      }
    ]
  }
}
```

## getSession

Gets an updated copy of the session object.

The scanner responds immediately to this command. If it cannot respond immediately for any reason, and it's available for use, it replies using the *waking* status, and the client must try again.

Clients should use `waitForEvents` to monitor for changes in the state of the scanner. However, in the event that the scanner reports `invalidState`, the client may use this command to discover the current state of the scanner.

### Session State Transitions

Current State	New State	Transition Notes
capturing	capturing	Remain in <i>capturing</i> .
closed	closed	Remain in <i>closed</i> .
draining	draining	Remain in <i>draining</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Remain in <i>ready</i> .

### Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
X-Privet-Token: token from /privet/info command<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "getSession",
  "params": {
    "sessionId": "sessionId from /privet/session/createSession"
  }
}
```

## Response

```
HTTP/1.1 200 OK<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "createSession",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    }
  }
}
```

## sendTask

Sends a TWAIN Direct task to the scanner.

The scanner responds immediately to this command. If it cannot respond immediately for any reason, and it's available for use, it replies using the *waking* status, and the client must try again.

Standard TWAIN Direct tasks may only be submitted while the scanner is in the *ready* state.

Vendor specific tasks are not required to have this limitation. Vendor specific actions may return a status of *commandPending*, indicating that the scanner is working on the request. The client is informed of the command's completion through `waitForEvents`. The event includes the `commandId` of the original `sendTask` command.

### params.task

A string. A TWAIN Direct Task, sent from the client to the scanner.

### results.session.task

A string. A complete TWAIN Direct Task, sent from the scanner to the client, reflecting the parts of the original task that the scanner was able to recognize and process.

## Session State Transitions

Current State	New State	Notes
capturing	capturing	Remain in <i>capturing</i> . Entire task must be vendor specific.
closed	closed	Remain in <i>closed</i> . Entire task must be vendor specific.
draining	draining	Remain in <i>draining</i> . Entire task must be vendor specific.
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Remain in <i>ready</i> .

## Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
X-Privet-Token: token from /privet/infoex command<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "sendTask",
  "params": {
    "sessionId": "sessionId from /privet/session/createSession",
    "task": { TWAIN Direct task }
  }
}
```

## Response

```
HTTP/1.1 200 OK<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "sendTask",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "Current scanner state",
      "task": { TWAIN Direct task reply }
    }
  }
}
```

## startCapturing

This command enables the scanner to scan sheets of paper.

Scanners vendors are strongly encouraged to provide some method of user proximity detection, even if it's only a button press on the physical scanner, before actually scanning paper.

### State Change Transitions

Current State	New State	Notes
capturing	capturing	Status is <i>invalidState</i> , remain in <i>capturing</i> .
closed	closed	Status is <i>invalidState</i> , remain in <i>closed</i> .
draining	draining	Status is <i>invalidState</i> , remain in <i>draining</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	capturing	Transition to <i>capturing</i> .

### Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
X-Privet-Token: token from /privet/info command<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "startCapturing",
  "params": {
    "sessionId": "sessionId from /privet/session/createSession"
  }
}
```

## Response

```
HTTP/1.1 200 OK<crif>
Content-Type: application/json; charset=UTF-8<crif>
Content-Length: number of bytes of UTF-8 encoded JSON data<crif>
<crif>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "startCapturing",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "capturing"
    }
  }
}
```

## readImageBlockMetadata

Returns information and attributes about the image, which may include non-image data contained within the image (such as barcode information).

### params.imageBlockNum

An integer number. It specifies the image block to be read.

### params.withThumbnail

A boolean. If true, the scanner includes a thumbnail of the image as part of the response.

### results.metadata

An object. TWAIN Direct metadata for a requested imageBlockNum.

## State Change Transitions

Current State	New State	Notes
capturing	capturing	Remain in <i>capturing</i> .
closed	closed	Remain in <i>closed</i> .
draining	draining	Remain in <i>draining</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Status is <i>invalidState</i> , remain in <i>ready</i> .

## Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
X-Privet-Token: token from /privet/infoex command<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "readImageBlockMetadata"
  "params": {
    "sessionId": "sessionId from /privet/session/createSession",
    "withThumbnail": false or true
  }
}
```

## Response without Thumbnail

```
HTTP/1.1 200 OK<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "readImageBlockMetadata",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    },
    "metadata": { Metadata for this image }
  }
}
```

## Response with Thumbnail

Pay special attention to all of the empty lines in this response

```
HTTP/1.1 200 OK<crf>
Content-Type: multipart/mixed; boundary="boundary-string"<crf>
Content-Length: number of bytes in response<crf>
<crf>
--boundary-string<crf>
Content-Type: application/json; charset=UTF-8<crf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crf>
<crf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "readImageBlockMetadata",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": "Current revision number for this session object",
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    },
    "metadata": { Metadata for this image }
  }
}<crf>
<crf>
--boundary-string<crf>
Content-Type: application/pdf<crf>
Content-Length: number of bytes of PDF/raster data<crf>
Content-Transfer-Encoding: binary<crf>
Content-Disposition: inline; filename="thumbnail.pdf"<crf>
<crf>
Thumbnail in PDF/raster format<crf>
<crf>
--boundary-string--<crf>
```

## readImageBlock

Transfer an imageBlock from the scanner to the client. An imageBlock contains a complete image or a fragment of an image. The metadata for the imageBlock indicates what it is.

### params.imageBlockNum

An integer number. It specifies the image block to be read.

### params.withMetadata

A boolean. If true, the scanner includes the metadata as part of the response. Use this property if skipping the call to readImageBlockMetadata.

### results.metadataTwainDirect

An object. TWAIN Direct metadata for a requested imageBlockNum. This is only returned if params.withMetadata was set to true.

## State Change on Success

Current State	New State	Notes
capturing	capturing	Remain in <i>capturing</i> .
closed	closed	Remain in <i>closed</i> .
draining	draining	Remain in <i>draining</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Status is <i>invalidState</i> , remain in <i>ready</i> .

## Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
X-Privet-Token: token from /privet/info command<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "readImageBlock"
  "params": {
    "sessionId": "sessionId from /privet/session/createSession",
    "withMetadata": false or true
  }
}
```

## Response

Pay special attention to all of the empty lines in this response

```
HTTP/1.1 200 OK<crlf>
Content-Type: multipart/mixed; boundary="boundary-string"<crlf>
Content-Length: number of bytes in response<crlf>
<crlf>
--boundary-string<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "readImageBlock",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": "Current revision number for this session object",
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    },
    "metadata": { Metadata for this image, if withMetadata is true }
  }
}<crlf>
<crlf>
--boundary-string<crlf>
Content-Type: application/pdf<crlf>
Content-Length: number of bytes of PDF/raster data<crlf>
Content-Transfer-Encoding: binary<crlf>
Content-Disposition: inline; filename="image.pdf"<crlf>
<crlf>
Image in PDF/raster format<crlf>
<crlf>
--boundary-string--<crlf>
```

## releaseImageBlocks

Release one or more imageBlocks in a contiguous range of imageBlock numbers.

A client may specify a value of 1 for the imageBlockNum, and the maximum 32-bit positive integer value (2147483647) for lastImageBlockNum, if its intention is to release all of the imageBlocks.

### params.imageBlockNum

An integer number. It specifies the first image to be deleted from the imageBlock.

### params.lastImageBlockNum

An integer number. It specifies the first image to be deleted from the imageBlock. If only one imageBlock is being released, then this number is the same as the params.imageBlockNum number.

## Session State

Current State	New State	Notes
capturing	capturing	Remain in <i>capturing</i> .
closed	closed	Remain in <i>closed</i> , if there are pending imageBlocks.
closed	noSession	Transition to <i>noSession</i> , if there are no pending imageBlocks.
draining	draining	Remain in <i>draining</i> , if there are pending imageBlocks.
draining	ready	Transition to <i>ready</i> , if there are no pending imageBlocks.
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Status is <i>invalidState</i> , remain in <i>ready</i> .

## Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
X-Privet-Token: token from /privet/infoex command<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "releaseImageBlocks"
  "params": {
    "sessionId": "sessionId from /privet/session/createSession",
    "imageBlockNum": number of first block to remove,
    "lastImageBlockNum": number of last block to remove
  }
}
```

## Response

```
HTTP/1.1 200 OK<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "releaseImageBlocks",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    },
    "metadata": { Metadata for this image }
  }
}
```

## stopCapturing

Gracefully stops the scanner from capturing more sheets (any sheet currently being scanned is completed). Images that have not been transferred remain as numbered imageBlocks in the session object.

After this call the client can change the scanner configuration with a call to sendTask. Scanning can be restarted with a call to startCapturing. The session can be ended with a call to closeSession.

### Session State Transitions

Current State	New State	Notes
capturing	draining	Transition to <i>draining</i> , if there are pending imageBlocks.
capturing	ready	Transition to <i>ready</i> , if there are no pending imageBlocks.
closed	closed	Status is <i>invalidState</i> , remain in <i>closed</i> .
draining	draining	Status is <i>invalidState</i> , remain in <i>draining</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Status is <i>invalidState</i> , remain in <i>ready</i> .

### Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crLf>
Content-Type: application/json; charset=UTF-8<crLf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crLf>
X-Privet-Token: token from /privet/info command<crLf>
<crLf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "stopCapturing"
  "params": {
    "sessionId": "sessionId from /privet/session/createSession"
  }
}
```

## Response

```
HTTP/1.1 200 OK<crif>
Content-Type: application/json; charset=UTF-8<crif>
Content-Length: number of bytes of UTF-8 encoded JSON data<crif>
<crif>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "stopCapturing",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    }
  }
}
```

## closeSession

No more sheets of paper can be scanned in this session. If there are no pending imageBlocks, the session ends.

If the scanner supports it, any client may create a new session while the current client finishes transferring and releasing pending imageBlocks for its session. Otherwise calls to createSession must be refused.

If the scanner is in a capturing state, and stopCapturing has not been called, then the current sheet of paper being scanned is aborted without transferring the rest of its data. An automatic document feeder ejects its current sheet. A flatbed scanner sends its camera to the home position. Clients must only use closeSession this way, when they want to exit quickly without transferring more images. Client that do this must also release all pending image blocks to finish closing the session.

### Session State Transitions

Current State	New State	Notes
capturing	capturing	Transition to <i>closed</i> , if there are pending imageBlocks.
capturing	noSession	Transition to <i>noSession</i> , if there are no pending imageBlocks.
closed	closed	Status is <i>invalidState</i> , remain in <i>closed</i> .
draining	draining	Transition to <i>closed</i> .
noSession	noSession	Status is <i>invalidState</i> , remain in <i>noSession</i> .
ready	ready	Status is <i>invalidState</i> , remain in <i>ready</i> .

## Command

```
POST <address>/privet/twaindirect/session HTTP/1.1<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
X-Privet-Token: token from /privet/infoex command<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "closeSession"
  "params": {
    "sessionId": "sessionId from /privet/session/createSession"
  }
}
```

## Response

```
HTTP/1.1 200 OK<crlf>
Content-Type: application/json; charset=UTF-8<crlf>
Content-Length: number of bytes of UTF-8 encoded JSON data<crlf>
<crlf>
{
  "kind": "twainlocalscanner",
  "commandId": "Client generated id for this command",
  "method": "closeSession",
  "results": {
    "success": true,
    "session": {
      "sessionId": "sessionId from /privet/session/createSession",
      "revision": Current revision number for this session object,
      "state": "Current scanner state",
      "imageBlocks": [ pending image blocks (if any) ]
    }
  }
}
```

---

## Sample Sessions

### Scanning a single sheet, duplex, bw1, 150 dpi, no compression

#### infoex

Get information about our scanner, and collect the x-privet-token we need for all of the other calls.

Session State: noSession

-----  
GET https://scanner.local:55555/privet/infoex HTTP/1.1  
X-Privet-Token: ""

-----  
HTTP/1.1 200 OK  
Content-Length: 447  
Content-Type: application/json; charset=UTF-8

```
{"version":"1.0","name":"scanner","description":"MyScanner","url":"https://scanner.local:55555/twaindirect","type":"twaindirect","id":"","device_state":"idle","connection_state":"offline","manufacturer":"","model":"","serial_number":"","firmware":"","uptime":"","setup_url":"","support_url":"","update_url":"","x-privet-token":"KJ412RS22PrNgb1leLDW8LHgcrk=:636167273550000965","api":["/privet/twaindirect/session"],"semantic_state":"","clouds":[]}
```

#### createSession

Create a session for ourselves, which locks the scanner transport for our exclusive use.

Session State: noSession --> ready

-----  
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1  
Content-Type: application/json; charset=UTF-8  
Content-Length: 104  
X-Privet-Token: KJ412RS22PrNgb1leLDW8LHgcrk=:636167273550000965

```
{"kind":"twainlocalscanner","commandId":"c291bcc0-94f5-4d83-88f9-947834916acb","method":"createSession"}
```

-----  
HTTP/1.1 200 OK  
Content-Type: application/json; charset=UTF-8  
Content-Length: 223

```
{"kind":"twainlocalscanner","commandId":"c291bcc0-94f5-4d83-88f9-947834916acb","method":"createSession","results":{"success":true,"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":1,"state":"ready"}}}
```

## **waitForEvents (request)**

Wait for an event from the scanner.

Session State: ready

---

```
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1
Content-Type: application/json; charset=UTF-8
Content-Length: 210
X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965
```

```
{"kind": "twainlocalscanner", "commandId": "a53e7081-2725-4854-ad0c-a5ef3d9fb079", "method": "waitForEvents", "params": {"sessionId": "c64a93ef-9855-4914-b295-7c245ca16466", "sessionRevision": 1}}
```

---

There is no response until the session object is updated outside of the bounds of a command. For instance, the next command updates the session object, but an event is not generated, since the response to the command contains the pertinent data.

---

## **sendTask**

Send a task with a configure action, to set up the scanner the way we want.

Session State: ready

---

```
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1
Content-Type: application/json; charset=UTF-8
Content-Length: 482
X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965
```

```
{"kind": "twainlocalscanner", "commandId": "f963de1f-8314-48b4-abc9-0417af0337e8", "method": "sendTask", "params": {"sessionId": "c64a93ef-9855-4914-b295-7c245ca16466", "task": {"actions": [{"action": "configure", "streams": [{"sources": [{"source": "any", "pixelFormats": [{"pixelFormat": "bw1", "attributes": [{"attribute": "compression", "values": [{"value": "none"}]}, {"attribute": "resolution", "values": [{"value": 150}, {"value": 200}], {"attribute": "numberOfSheets", "values": [{"value": 1}]}]}]}]}]}]}
```

---

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 567
```

```
{"kind": "twainlocalscanner", "commandId": "f963de1f-8314-48b4-abc9-0417af0337e8", "method": "sendTask", "results": {"success": true, "session": {"sessionId": "c64a93ef-9855-4914-b295-7c245ca16466", "revision": 2, "state": "ready", "task": {"actions": [{"action": "configure", "results": {"success": true}, "streams": [{"stream": "stream0", "sources": [{"source": "any", "pixelFormats": [{"pixelFormat": "bw1", "attributes": [{"attribute": "compression", "values": [{"value": "none"}]}, {"attribute": "resolution", "values": [{"value": 150}], {"attribute": "numberOfSheets", "values": [{"value": 1}]}]}]}]}]}]}
```

---

### **startCapturing**

Tell the scanner to start capturing sheets of paper.

Session State: ready --> capturing

---

POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1

Content-Type: application/json; charset=UTF-8

Content-Length: 167

X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965

```
{"kind":"twainlocalscanner","commandId":"2405d9aa-5088-476e-859e-c2b49d01360e","method":"startCapturing", "params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466"}}
```

---

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

Content-Length: 228

```
{"kind":"twainlocalscanner","commandId":"2405d9aa-5088-476e-859e-c2b49d01360e","method":"startCapturing", "results":{"success":true,"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":3,"state":"capturing"}}
```

### **waitForEvents (response)**

Event received.

Session State: capturing

---

This is the response to the waitForEvents call made previously.

---

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

Content-Length: 228

```
{"kind":"twainlocalscanner","commandId":"a53e7081-2725-4854-ad0c-a5ef3d9fb079","method":"waitForEvents", "results":{"success":true,"events":[{"event":"imageBlocks","session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":4,"state":"capturing","imageBlocksDrained":false,"imageBlocks":[1]}]}}
```

### **waitForEvents (request)**

Wait for an event from the scanner.

Session State: capturing

-----  
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1

Content-Type: application/json; charset=UTF-8

Content-Length: 210

X-Privet-Token: KJ412RS22PrNgbl1eLDW8LHgcrk=:636167273550000965

```
{"kind":"twainlocalscanner","commandId":"b200b814-03aa-4d71-a24b-f77ab8137d37","method":"waitForEvents", "params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","sessionRevision":1}}
```

-----  
There is no immediate response.  
-----

## **readImageBlock**

Get image and metadata in the same call, this reads the front of a sheet of paper.

Session State: capturing

-----  
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1

Content-Type: application/json; charset=UTF-8

Content-Length: 166

X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965

```
{"kind":"twainlocalscanner","commandId":"39c39eb2-7903-4324-ba9f-41d07bde5b2a","method":"readImageBlock","params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","withMetadata":true,"imageBlockNum":1}}
```

-----  
HTTP/1.1 200 OK

Content-Type: multipart/mixed; boundary="WaFfLeSaReTaStY"

Content-Length: 266025

--WaFfLeSaReTaStY

Content-Type: application/json; charset=UTF-8

Content-Length: 605

```
{"kind":"twainlocalscanner","commandId":"39c39eb2-7903-4324-ba9f-41d07bde5b2a","method":"readImageBlock","results":{"success":true,"metadata":{"address":{"imageNumber":1,"sheetNumber":1,"source":"feederFront"},"image":{"compression":"none","pixelFormat":"bw1","pixelHeight":1650,"pixelOffsetX":0,"pixelOffsetY":0,"pixelWidth":1280,"resolution":150},"imageBlock":{"imageNumber":1,"imagePart":1,"moreParts":false},"status":{"success":true}}},"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":5,"state":"capturing","doneCapturing":false,"imageBlocksDrained":false,"imageBlocks":[1,2]}}
```

--WaFfLeSaReTaStY

Content-Type: application/pdf

Content-Length: number of bytes of PDF/raster data

Content-Transfer-Encoding: binary

Content-Disposition: inline; filename="image.pdf"

Image in PDF/raster format

--WaFfLeSaReTaStY--  
-----

## **releaseImageBlocks**

Always release a transferred image as quickly as possible.

Session State: capturing

-----  
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1

Content-Type: application/json; charset=UTF-8

Content-Length: 211

X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965

```
{"kind":"twainlocalscanner","commandId":"c51a490f-cc34-45ae-afb4-fd9eae0cb844","method":"releaseImageBlocks","params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","imageBlockNum":1,"lastImageBlockNum":1}}
```

-----  
HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

Content-Length: 277

```
{"kind":"twainlocalscanner","commandId":"c51a490f-cc34-45ae-afb4-fd9eae0cb844","method":"releaseImageBlocks","results":{"success":true,"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":6,"state":"capturing","imageBlocksDrained":false,"imageBlocks":[2]}}
```

## **readImageBlock**

Get image and metadata in the same call, this reads the rear of a sheet of paper.

Session State: capturing

-----  
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1

Content-Type: application/json; charset=UTF-8

Content-Length: 205

X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965

```
{"kind":"twainlocalscanner","commandId":"800ae40e-1fb1-486a-a00b-253ec7607fd7","method":"readImageBlock","params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","withMetadata":true,"imageBlockNum":2}}
```

-----  
HTTP/1.1 200 OK

Content-Type: multipart/mixed; boundary="WaFfLeSaReTaStY"

Content-Length: 265995

--WaFfLeSaReTaStY

Content-Type: application/json; charset=UTF-8

Content-Length: 603

```
{"kind":"twainlocalscanner","commandId":"800ae40e-1fb1-486a-a00b-253ec7607fd7","method":"readImageBlock","results":{"success":true,"metadata":{"address":{"imageNumber":2,"sheetNumber":1,"source":"feederRear"},"image":{"compression":"none","pixelFormat":"bw1","pixelHeight":1650,"pixelOffsetX":0,"pixelOffsetY":0,"pixelWidth":1280,"resolution":150},"imageBlock":{"imageNumber":2,"imagePart":1,"moreParts":false},"status":{"success":true}},"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":6,"state":"capturing","doneCapturing":false,"imageBlocksDrained":false,"imageBlocks":[2]}}
```

--WaFfLeSaReTaStY

Content-Type: application/pdf

Content-Length: number of bytes of PDF/raster data

Content-Transfer-Encoding: binary

Content-Disposition: inline; filename="image.pdf"

Image in PDF/raster format

--WaFfLeSaReTaStY--  
-----

## **releaseImageBlocks**

Always release a transferred image as quickly as possible.

Session State: capturing

---

```
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1
Content-Type: application/json; charset=UTF-8
Content-Length: 211
X-Privet-Token: KJ412RS22PrNgbl1eLDW8LHgrk=:636167273550000965
```

```
{"kind":"twainlocalscanner","commandId":"589cb8dc-efa0-4095-92eb-1bb5f4a09f7b","method":"releaseImageBl
ocks","params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","imageBlockNum":2,"lastImageBlockN
um":2}}
```

---

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 275
```

```
{"kind":"twainlocalscanner","commandId":"589cb8dc-efa0-4095-92eb-1bb5f4a09f7b","method":"releaseImageBl
ocks","results":{"success":true,"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":7,"st
ate":"capturing","imageBlocksDrained":true,"imageBlocks":[]}}
```

## **stopCapturing**

Tell the scanner that we're done capturing images.

Session State: capturing --> ready

---

```
POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1
Content-Type: application/json; charset=UTF-8
Content-Length: 166
X-Privet-Token: KJ412RS22PrNgbl1eLDW8LHgrk=:636167273550000965
```

```
{"kind":"twainlocalscanner","commandId":"bac6b488-58de-4fd7-bb2e-6a3ac1355006","method":"stopCapturing
","params":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466"}}
```

---

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 223
```

```
{"kind":"twainlocalscanner","commandId":"bac6b488-58de-4fd7-bb2e-6a3ac1355006","method":"stopCapturing
","results":{"success":true,"session":{"sessionId":"c64a93ef-9855-4914-b295-7c245ca16466","revision":8,"state":
"ready"}}
```

## **closeSession**

End the session.

Session State: ready --> noSession

---

POST https://scanner.local:55555/privet/twaindirect/session HTTP/1.1

Content-Type: application/json; charset=UTF-8

Content-Length: 165

X-Privet-Token: KJ412RS22PrNgb1eLDW8LHgcrk=:636167273550000965

```
{"kind": "twainlocalscanner", "commandId": "02b2d221-af8e-445f-86c5-584bd53c6c4f", "method": "closeSession", "params": {"sessionId": "c64a93ef-9855-4914-b295-7c245ca16466"}}
```

---

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

Content-Length: 223

```
{"kind": "twainlocalscanner", "commandId": "02b2d221-af8e-445f-86c5-584bd53c6c4f", "method": "closeSession", "results": {"success": true, "session": {"sessionId": "c64a93ef-9855-4914-b295-7c245ca16466", "revision": 9, "state": "closed"}}
```