TWAIN Direct™ Specification: Task

(DRAFT COPY)

February 22nd 2017
Revision 0.24

This is a draft copy of the proposed TWAIN Direct Specification: Task Configure. Its contents may be added to, changed or deleted at any time.

History

Date	Version	Comment
July 1st, 2014	0.01	Initial draft.
December 4th, 2014	0.02	Revision after comments (folded in the contents of the Introduction to TWAIN Direct)
February 3rd, 2015	0.03	Revisions made as part of the February Tech meeting
February 10th, 2015	0.04	Added Rebecca's comments, finish resolving all comments
March 3rd, 2015	0.05	Added new metadata content
March 17th, 2015	0.06	Comments from Mihail, metadata changes
March 21st, 2015	0.06	Expanded the section on Exceptions
April 21st, 2015	0.07	Prep work for releasing draft copy
April 21st, 2015	0.08	Release of draft copy
April 29th, 2015	0.09	Addition of TWAIN Direct Task Reply and Using the Spec
June 13th, 2015	0.10	Cleanup pass of the user guide portion of the Spec, added comment property, added exception and vendor to the value object
July 3rd, 2015	0.11	Added note about JSON types and 64-bit numbers
June 30th, 2015	0.12	Started appendix on certification
August 8th, 2015	0.13	Wordsmithing, fixes for exception and error handling sections
August 27th, 2015	0.14	Fixed some formatting issues
September 8th, 2015	0.15	Finally transferred the caps and metadata from TWAIN, added cross references within the document
December 3rd, 2015	0.16	Comments during the quarterly meeting
December 17th, 2015	0.17	More comments from the quarterly meeting
January 18th, 2016	0.18	Updated for client/scanner and virtual scanner discussions
June 9th, 2016	0.19	Split into its own document
September 12th, 2016	0.20	Added "stream" to "streams"
November 28th, 2016	0.21	Changed the title
January, 13th, 2017	0.22	Adding new actions
February 8th, 2017	0.23	Work on the4 encryption section, fixed the vendor properties
February 22nd, 2017	0.24	Added name properties

Notes

Notes
• The User Guide is done, please review (Title Page through PDF/raster) • The Metadata Chapter is done, please review • The Topology Chapters are done, please review • The Attributes Chapter is done, please review • Appendix A needs some more samples and a link to twaindirect.org • Appendix B is waiting for the NDA, and needs some additional work • Appendix C needs work

Contents

History

Notes

Contents

Glossary of Terms

References

Task and Action Objects
- task
- task.actions[ ]
- task.actions[ ].action
- task.actions[ ].comment
- task.actions[ ].exception
- task.actions[ ].vendor

Action: configure
- Stream Object
  - task.actions[ ].streams[ ]
  - task.actions[ ].streams[ ].comment
  - task.actions[ ].streams[ ].exception
  - task.actions[ ].streams[ ].name
  - task.actions[ ].streams[ ].vendor
- Source Object
  - task.actions[ ].streams[ ].sources[ ]
  - task.actions[ ].streams[ ].sources[ ].comment
  - task.actions[ ].streams[ ].sources[ ].exception
  - task.actions[ ].streams[ ].sources[ ].name
  - task.actions[ ].streams[ ].sources[ ].source
  - task.actions[ ].streams[ ].sources[ ].vendor
- PixelFormat Object
  - task.actions[ ].streams[ ].sources[ ].pixelFormats[ ]
  - task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].comment
  - task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].exception

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].name
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].pixelFormat
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].vendor
Attribute Object
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ]
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].attribute
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].comment
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].exception
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].vendor
Value Object
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ]
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].comment
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].exception
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].value
task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].vendor

Action: encryptionProfiles
encryptionProfile Object
task.actions[ ].encryptionProfiles[ ]
task.actions[ ].encryptionProfiles[ ].comment
task.actions[ ].encryptionProfiles[ ].exception
task.actions[ ].encryptionProfiles[ ].profile
task.actions[ ].encryptionProfiles[ ].vendor

Action: encryptionPublicKeys
encryptionPublicKeys Object
task.actions[ ].encryptionPublicKeys[ ]
task.actions[ ].encryptionPublicKeys[ ].base64PublicKey
task.actions[ ].encryptionPublicKeys[ ].comment
task.actions[ ].encryptionPublicKeys[ ].exception
task.actions[ ].encryptionPublicKeys[ ].publicKeyType
task.actions[ ].encryptionPublicKeys[ ].vendor

Action: encryptionReport
encryptionReport Object
task.actions[ ].encryptionReport
task.actions[ ].encryptionReport.digitalSignatures
task.actions[ ].encryptionReport.digitalSignatures[ ].comment

task.actions[ ].encryptionReport.digitalSignatures[ ].digitalSignature
task.actions[ ].encryptionReport.digitalSignatures[ ].vendor
task.actions[ ].encryptionReport.encryptionProfiles
task.actions[ ].encryptionReport.encryptionProfiles[ ].comment
task.actions[ ].encryptionReport.encryptionProfiles[ ].profile
task.actions[ ].encryptionReport.encryptionProfiles[ ].vendor
task.actions[ ].encryptionReport.encryptionPublicKeys
task.actions[ ].encryptionReport.encryptionPublicKeys[ ].comment
task.actions[ ].encryptionReport.encryptionPublicKeys[ ].vendor

TWAIN Direct Attributes
- Overview
- alarms
- alarmVolume
- automaticDeskew
- automaticSize
- barcodes
- bitDepthReduction
- brightness
- compression
- continuousScan
- contrast
- cropping
- discardBlankImages
- doubleFeedDetection
- doubleFeedDetectionLength
- doubleFeedDetectionResponse
- doubleFeedDetectionSensitivity
- flipRotation
- height
- imageMerge
- imageMergeHeightThreshold
- invert
- jpegQuality
- micr
- mirror
- noiseFilter

overScan
numberOfSheets
offsetX
offsetY
patchCodes
resolution
rotation
sheetHandling
sheetSize
threshold
uncalibratedImage
width

Glossary of Terms

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

Word	Meaning
action	A TWAIN Direct command (e.g. “configure”).
application	A program that sends TWAIN Direct commands to a scanner.
attribute	A configurable item, such as compression, resolution, etc.
communication manager	A system that discovers scanners, registers them and provides cloud and/or local area net communication channels.
exception	A TWAIN Direct directive that changes the way a TWAIN Direct task is evaluated by a scanner, when it cannot exactly match a specific request within a task.
JSON	A lightweight data-interchange format.
pixelFormat	The combination of a color space and a bit depth, for instance, rgb24 indicates a color image with 24 bits of depth.
scanner	Any device that captures images for an application.
source	A physical provider of images, such as a flatbed or an automatic document feeder.
stream	A collection of one or more sources, which combined together results in a stream of images during scanning.
task	A TWAIN Direct construct used to issue actions to a scanner.
topology	The combination in a configure action of a stream, source and pixelFormat used to address components within the scanner.
user	A person in control of an application and a scanner.

References

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

Word	Meaning
Base64	Refer to 5.2 Base64 Content-Transfer-Encoding http://www.w3.org/Protocols/rfc1341/5_Content-Transfer-Encoding.html
Google JSON Style Guide	Google JSON Style Guide https://google-styleguide.googlecode.com/svn/trunk/jsoncstyleguide.xml
JavaScript Reserved Words	List of reserved words http://www.w3schools.com/js/js_reserved.asp
JSON	RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON) https://www.ietf.org/rfc/rfc4627.txt TWAIN Direct requires all task content to be contained in an object token. With this restriction in place the following JSON parsers may also be used. Just confirm that the outmost token is an object before proceeding. ECMA-404 - http://www.json.org RFC 7159 - http://www.rfc-editor.org/rfc/rfc7159.txt A convenient tool to compact, beautify, and validate JSON data https://jsonformatter.curiousconcept.com/
PDF/raster	PDF Raster Documents http://pdfraster.org
TWAIN DIrect Sample Code	Website for TWAIN Direct sample code **TBD**
TWAIN Direct UUID Version 1	**211a1e90-11e1-11e5-9493-1697f925ec7b** https://www.uuidgenerator.net (generation source)
UUID	A Universally Unique IDentifier (UUID) URN Namespace Per the 2008-08 spec UUIDs must be generated with lowercase letters, but code compaging UUIDs must be insensitive to case. http://www.ietf.org/rfc/rfc4122.txt http://www.itu.int/rec/T-REC-X.667/en
TWAIN Direct	Website for TWAIN Direct http://twaindirect.org

Task and Action Objects

task

Description The outermost TWAIN Direct object.

Presence Mandatory.

If the object is empty the scanner returns success, but takes no other action. The task is complete after the status is returned.

Members actions

Examples

```
{
  ...
}
```

task.actions[ ]

Description An array of one or more action objects for this task.

Presence Optional. If not present the scanner returns success, but takes no other action. The task is complete after the status is returned.

Mandatory members are marked with a one ([1]), and must be included if the actions array is present.

Members
action[1]
comment
exception
streams
vendor

Examples

```
{
  "actions": [
    {
      ...
    }
  ]
}
```

task.actions[ ].action

Description	An action for the scanner to take.
Presence	Optional. If not present the scanner assumes the value is configure.
Values	String, one of the following.
configure	Configure the scan session. The task should include a streams array with at least one stream object in it.
encryptionProfiles	Select zero or more encryption profiles inside of the scanner for encrypting images.
encryptionPublicKeys	Set zero or more public keys for encrypting images.
reportFeatures	Request a feature report from the scanner.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      ...
    }
  ]
}
```

task.actions[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values

(any valid UTF8-encoded JSON string)

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "comment": "Product Acme",
      "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
      ...
    }
  ]
}
```

task.actions[ ].exception

Description A string. The exception to take if a problem occurs in the body of the current action.

Presence Optional. If not present each action in the array defaults to “nextObject”.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "exception": "fail",
      ...
    }
  ]
}
```

task.actions[ ].vendor

Description A string. The UUID of the vendor defining the action. If the scanner recognizes the UUID then it examines the action. If it does not recognize it, then it ignores the entire action.

Presence Optional. If not present the scanner defaults to the TWAIN Direct UUID, which is supported by all scanners.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "vendor": "c1528f4f-b6a2-46ca-a7b0-2c40be74a5ab",
      ...
    }
  ]
}
```

Action: configure

The configure action sets mechanical and image processing attributes prior to capturing images from sheets of paper.

Stream Object

task.actions[ ].streams[ ]

Description An array of one or more stream objects for a "configure" action. If the action isn't set to "configure" the object is ignored.

Presence Optional. If not present the scanner scans uses its default stream. This default is determined by the scanner vendor.

Members
comment
exception
name
sources
vendor

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "comment": "Product Acme",
          "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].exception

Description A string. The exception to take if a problem occurs in the body of the current stream.

Presence Optional. If not present the value is inherited from the action object, if one was explicitly set.

If a value was not set for the action object, then the default value is “nextStream” for all streams in the array, except the last one, which defaults to “ignore”.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged. The first stream with ignore set will always be selected, and any streams that follow it will never be used.
nextStream	Discard the current stream and proceed to the next stream in the array. If there is no next steam, this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "exception": "nextStream",
          ...
        },
        {
          "exception": "ignore",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].name

Description A string. A name for the stream. It has no intrinsic meaning, and is not required to be unique within a task. The value is included in the metadata for an image as streamName.

Presence Optional. If not present the scanner fills the property in the reply with a string of the form “stream#”, where # counts the streams under the current action (including vendor custom streams), starting from zero.

Values Any UTF-8 encoded JSON string. Scanners are only required to store the first 32 characters.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "name": "my stream",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].vendor

Description A string. The UUID of the vendor defining the stream. If the scanner recognizes the UUID then it examines the stream. If it does not recognize it, then it ignores the entire stream.

Presence Optional. If not present the value is inherited from the action object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "comment": "Product Acme",
          "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
          ...
        }
      ]
    }
  ]
}
```

Source Object

task.actions[ ].streams[ ].sources[ ]

Description An array of one or more source objects for this stream.

Presence Optional. If not present the scanner scans using its default settings.

Members
comment
exception
name
pixelFormats
source
vendor

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              ...
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "comment": "Product Acme",
              "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
              ...
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].exception

Description A string. The exception to take if a problem occurs in the body of the current source.

Presence Optional. If not present the value is inherited from the stream object.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged. The first stream with ignore set will always be selected, and any streams that follow it will never be used.
nextStream	Discard the current stream and proceed to the next stream in the array. If there is no next steam, then this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "exception": "fail",
              ...
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].name

Description A string. A name for the source. It has no intrinsic meaning, and is not required to be unique within a task. The value is included in the metadata for an image as sourceName.

Presence Optional. If not present the scanner fills the property in the reply with a string of the form “source#”, where # counts the source under the current stream (including vendor custom sources), starting from zero.

Values Any UTF-8 encoded JSON string. Scanners are only required to store the first 32 characters.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "name": "my source",
              ...
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].source

Description A string that designates a distinct *source* of images within a scanner, such as the flatbed or ADF. Standard values are listed below.

Presence Optional. If not present, defaults to “any”.

Values

any	Any source at the scanner’s discretion, this includes custom sources that are not defined by the TWAIN Direct specification.
feeder	An automatic document feeder. This can be used with scanners that scan just one side or both sides of a sheet of paper.
feederFront	The part of an automatic document feeder that scans the front of each sheet of paper. If a scanner does not support independent control of feederFront and feederRear sources, then it uses the feederFront source, and the feederRear source is ignored.
feederRear	The part of an automatic document feeder that scans the rear of each sheet of paper. If a scanner does not support independent control of feederFront and feederRear sources, then it uses feederRear if and only if it’s the only source specified in the stream.
flatBed	A glass surface that the paper is set upon.
planetary	A mounted camera, typically used for scanning books.
storage	An existing repository of images.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
```

```
        {
          "sources": [
            {
              "source": "feeder",
              ...
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].vendor

Description A string. The UUID of the owner of the source. If the scanner recognizes the UUID then it examines the source. If it does not recognize it, then it ignores the entire source.

Presence Optional. If not present the value is inherited from the stream object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "comment": "Product Acme",
              "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
              ...
            }
          ]
        }
      ]
    }
  ]
}
```

PixelFormat Object

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ]

Description An array of one or more pixelFormat objects for this source.

Presence Optional. If not present the scanner scans using its default settings.

Members
attributes
comment
exception
name
pixelFormat
vendor

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  ...
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "comment": "Product Acme",
                  "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
                  ...
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].exception

Description A string. The exception to take if a problem occurs in the body of the current pixelFormat.

Presence Optional. If not present the value is inherited from the source object.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged. The first stream with ignore set will always be selected, and any streams that follow it will never be used.
nextStream	Discard the current stream and proceed to the next stream in the array. If there is no next steam, then this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "exception": "fail",
                  ...
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].name

Description A string. A name for the pixelFormat. It has no intrinsic meaning, and is not required to be unique within a task. The value is included in the metadata for an image as pixelFormatName.

Presence Optional. If not present the scanner fills the property in the reply with a string of the form “pixelFormat#”, where # counts the pixelFormat under the current source (including vendor custom pixelFormats), starting from zero.

Values Any UTF-8 encoded JSON string. Scanners are only required to store the first 32 characters.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "name": "my pixelFormat",
                  ...
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].pixelFormat

Description A string. The colorspace and the bit depth of a pixel.

Presence Optional. If not present the default is at the discretion of the scanner.

Values

bw1	Black-and-white with a bit depth of 1, also called packed bitonal, since an 8-bit byte contains 8 of these pixelFormats.
gray8	Grayscale with a bit depth of 8, allowing for 256 shades of grey.
gray16	Grayscale with a bit depth of 16, allowing for 65536 shades of grey.
rgb24	Color with a bit depth of 24 (8-bits per channel), allowing for 16.7 million colors.
rgb48	Color with a bit depth of 48 (16-bits per channel), allowing for 281 trillion colors.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "pixelFormat": "rgb24",
                  ...
                }
              ]
            }
          ]
        }
      ]
    }
```

```
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].vendor

Description A string. The UUID of the owner of the pixelFormat. If the scanner recognizes the UUID then it examines the pixelFormat. If it does not recognize it, then it ignores the entire pixelFormat.

Presence Optional. If not present the value is inherited from the source object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "comment": "Product Acme",
                  "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
                  ...
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Attribute Object

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ]

Description An array of one or more attribute objects for this pixelFormat.

Presence Optional. If not present the scanner scans using its default settings.

Members attribute
comment
exception
values
vendor

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      ...
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].attribute

Description A string. The colorspace and the bit depth of a pixel.

Presence Optional. If not present the default is at the discretion of the scanner.

Values

Refer to the section on TWAIN Direct Attributes for the complete list

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "compression",
                      ...
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "comment": "Product Acme",
                      "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
                      ...
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].exception

Description A string. The exception to take if a problem occurs in the body of the current attribute.

Presence Optional. If not present the value is inherited from the pixelFormat object.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged. The first stream with ignore set will always be selected, and any streams that follow it will never be used.
nextStream	Discard the current stream and proceed to the next stream in the array. If there is no next steam, then this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "exception": "fail",
                      ...
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
```

```
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].vendor

Description A string. The UUID of the owner of the attribute. If the scanner recognizes the UUID then it examines the attribute. If it does not recognize it, then it ignores the entire attribute.

Presence Optional. If not present the value is inherited from the pixelFormat object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "comment": "Product Acme",
                      "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
                      ...
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Value Object

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ]

Description An array of one or more value objects for this attribute. The scanner selects the first value that it supports and ignores all others.

Presence Optional. If not present the scanner scans using its default settings.

Members comment
exception
value
vendor

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "values": [
                        {
                          ...
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "values": [
                        {
                          "comment": "Product Acme",
                          "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
                          ...
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].exception

Description A string. The exception to take if a problem occurs in the body of the current value.

Presence Optional. If not present the value is inherited from the attribute object.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged. The first stream with ignore set will always be selected, and any streams that follow it will never be used.
nextStream	Discard the current stream and proceed to the next stream in the array. If there is no next steam, then this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "values": [
                        {
                          "exception": "fail",
                          ...
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
```

```
            }
          ]
        }
    ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].value

Description A string. A proposed value for this attribute.

Presence Optional. If not present the scanner scans using its default settings.

Values

Refer to the section on TWAIN Direct Attributes for the complete list

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "values": [
                        {
                          "value": "some value"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

task.actions[ ].streams[ ].sources[ ].pixelFormats[ ].attributes[ ].values[ ].vendor

Description A string. The UUID of the owner of the value. If the scanner recognizes the UUID then it examines the value. If it does not recognize it, then it ignores the entire value.

Presence Optional. If not present the value is inherited from the attribute object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "values": [
                        {
                          "comment": "Product Acme",
                          "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
                          ...
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Action: encryptionProfiles

The encryptionProfiles action specifies zero or more profiles, which are used to encrypt images captured by the scanner associated with this task.

A profile exposes a name, but conveys no other information to a user about how images will be encrypted. This is by design. The vendor has complete control how a profile specifies encryption for an image.

The scanner vendor is responsible for managing encryption profiles in their scanner. The TWAIN Working Group offers no recommendation at this time for standardizing these actions, but may describe one at a future date.

A user selects a profile with a meaningful name to encrypt their images. It's recommended that the name reflect the intent of the encryption: either the kind of images being captured, or the application that receive them. For instance, useful profile names along the lines of: "Insurance Forms", or "Tax Application".

In TWAIN Direct all exceptions default to "ignore", so to guarantee that an encryption profile is used it should have its exception set to "fail".

encryptionProfiles and encryptionPublicKeys may be specified in the same task.

Encrypted images must be digitally signed by the scanner.

encryptionProfile Object

task.actions[ ].encryptionProfiles[ ]

Description An array of one or more encryptionProfile objects for an “encryptionProfiles” action. The images are encrypted using the supplied profiles.

Presence Optional. If not present or empty the current profile selections are cleared. The default is to deliver unencrypted images.

Members comment
exception
profile
vendor

Examples

```
{
  "actions": [
    {
      "action": "encryptionProfiles",
      "encryptionProfiles": [
        {
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionProfiles[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionProfiles",
      "encryptionProfiles": [
        {
          "comment": "a comment",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionProfiles[ ].exception

Description A string. The exception to take if a problem occurs in the body of the encryptionProfiles.

Presence Optional. If not present the value is inherited from the action object.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged.
nextAction	Discard the current action and proceed to the next action in the array. If there is no next action, this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "encryptionProfiles",
      "encryptionProfiles": [
        {
          "exception": "fail",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionProfiles[ ].profile

Description A string. The name of an encryptionProfile inside of the scanner, matching one of the encryption profiles returned by the reportFeatures action.

Presence Mandatory. If not present the exception is applied to the object.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionProfiles",
      "encryptionProfiles": [
        {
          "profile": "Alice's Profile"
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionProfiles[ ].vendor

Description A string. The UUID of the owner of the encryptionProfile. If the scanner recognizes the UUID it examines the encryptionProfile. If it does not recognize it, it ignores the encryptionProfile.

Presence Optional. If not present the value is inherited from the attribute object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "encryptionProfiles",
      "encryptionProfiles": [
        {
          "comment": "Product Acme",
          "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
          "profile": "Alice's Profile"
        }
      ]
    }
  ]
}
```

Action: encryptionPublicKeys

The encryptionPublicKeys action sends zero or more public keys to the scanner, which are used to encrypt captured images associated with this task. encryptionPublicKeys are discarded by the scanner when they task is complete. They are not saved by the scanner.

In TWAIN Direct all exceptions default to “ignore”, so to guarantee that an encryption public key is used it should have its exception set to “fail”.

encryptionProfiles and encryptionPublicKeys may be specified in the same task.

Encrypted images must be digitally signed by the scanner.

encryptionPublicKeys Object

task.actions[ ].encryptionPublicKeys[ ]

Description An array of one or more encryptionPublicKeys objects for an “encryptionPublicKeys” action.

Presence Optional. If not present or present but empty, all current public keys are deleted from the scanner.

Members base64PublicKey
comment
exception
publicKeyType
vendor

Examples

```
{
  "actions": [
    {
      "action": "encryptionPublicKeys",
      "encryptionPublicKeys": [
        {
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionPublicKeys[ ].base64PublicKey

Description A base64 encoded string. The public key that the scanner uses to encrypt images. When decoded back to a string, the publicKeyType property indicates the format of the data.

Presence Mandatory. If not present the exception is applied to the object.

Values Any valid base64 string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionPublicKeys",
      "encryptionPublicKeys": [
        {
          "base64PublicKey": "public key data in base64 format",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionPublicKeys[ ].comment

Description A string. Ignored by the scanner, it allows an application writer to include useful descriptive information within a task.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionPublicKeys",
      "encryptionPublicKeys": [
        {
          "comment": "a comment",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionPublicKeys[ ].exception

Description A string. The exception to take if a problem occurs in the body of the encryptionProfiles.

Presence Optional. If not present the value is inherited from the action object.

Values

fail	Reject the entire task when encountering an unrecognized or unsupported property or value.
ignore	Ignore unrecognized or unsupported properties or values leaving current values unchanged.
nextAction	Discard the current action and proceed to the next action in the array. If there is no next action, this is interpreted as “fail”.

Examples

```
{
  "actions": [
    {
      "action": "encryptionPublicKeys",
      "encryptionPublicKeys": [
        {
          "exception": "fail",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionPublicKeys[ ].publicKeyType

Description A string. Indicates the format of the data encoded as base64 in the base64PublicKey.

Presence Optional. If not present the default is pem.

Values

pem	When decoded, the string obtained from the base64PublicKey data is in the PEM file format.

Examples

```
{
  "actions": [
    {
      "action": "encryptionPublicKeys",
      "encryptionPublicKeys": [
        {
          "publicKeyType": "pem",
          ...
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionPublicKeys[ ].vendor

Description A string. The UUID of the owner of the encryptionPublicKey. If the scanner recognizes the UUID then it examines the encryptionPublicKey. If it does not recognize it, then it ignores the entire encryptionPublicKey.

Presence Optional. If not present the value is inherited from the action object.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "encryptionPublicKeys",
      "encryptionPublicKeys": [
        {
          "profile": "Alice's Profile",
          "comment": "Product Acme",
          "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
          ...
        }
      ]
    }
  ]
}
```

Action: encryptionReport

Use this action to discover if a scanner supports encryption and digital signatures for the images it captures. The action sent to the scanner takes the form:

```
{
  "actions": [
    {
      "action": "encryptionReport"
    }
  ]
}
```

Supported encryption features are returned in an encryptionReport object. If the encryptionReport action is not supported the scanner returns the action without the report or with an empty encryptionReport object.

encryptionReport Object

task.actions[ ].encryptionReport

Description An object. It contains the reports returned by the scanner.

Presence Optional. If not present or present but empty the action is reported back without the encryptionReport or with an empty encryptionReport object.

Members digitalSignatures
encryptionProfiles
encryptionPublicKeys

Examples

Unsupported.

```
{
  "actions": [
    {
      "action": "encryptionReport"
      ]
    }
  ]
}
```

Supported, but no data to return.

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        }
      ]
    }
  ]
}
```

Supported, and with data...

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
          ...
        }
      ]
    }
  ]
```

```
}
```

task.actions[ ].encryptionReport.digitalSignatures

Description An array of one or more digitalSignature objects.

Digital signatures are stored inside of the scanner. They are used to sign images, and are intended to give an application confidence that the data it receives has not been tampered with while in transit from the scanner.

Scanners are encouraged to support at least two digital signatures: one that identifies the scanner, and one that can be set by the user.

Presence Optional, but recommended if encryption is not supported.

Mandatory if the scanner is configured to return encrypted images using the encryptionProfile or encryptionPublicKeys actions.

Members comment
digitalSignature
vendor

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "digitalSignatures": [
          {
            ...
          }
        }
      ]
    }
  ]
}
```

task.actions[ ].encryptionReport.digitalSignatures[ ].comment

Description A string. Information that may be helpful for the user.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "digitalSignatures": [
          {
            "digitalSignature": "Scanner XYZ"
          },
          {
            "comment": "Must be renewed at the end of the year",
            "digitalSignature": "Acme"
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.digitalSignatures[ ].digitalSignature

Description A string. A friendly name for a digital signature, such as “Scanner’s Signature” or “Company’s Signature”.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "digitalSignatures": [
          {
            "digitalSignature": "Scanner XYZ"
          },
          {
            "digitalSignature": "Acme"
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.digitalSignatures[ ].vendor

Description A string. The UUID of the owner of the digitalSignature. If the application recognizes the UUID then it may examine the rest of the object for content specific to that vendor.

Presence Optional.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "digitalSignatures": [
          {
            "digitalSignature": "Scanner XYZ"
          },
          {
            "comment": "Acme",
            "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
            "digitalSignature": "Acme",
            ...
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionProfiles

Description An array of zero or more encryptionProfile objects.

Encryption profiles are stored inside of the scanner. Scanner vendors are responsible for designing methods of

Presence Optional. If the array is not included in the report, the scanner does not support profiles. If the array is present but empty, the scanner supports profiles, but does not currently have any that it can use.

Members comment
profile
vendor

Examples

The scanner does not support profiles.

```
{
  "actions": [
    {
      "action": "encryptionReport"
      "encryptionReport": (
      }
    }
  ]
}
```

The scanner supports profile, but none are currently available.

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": (
        "encryptionProfiles": [
        ]
      }
    }
  ]
}
```

The scanner supports profiles, and two are currently available.

```
{
  "actions": [
    {
```

```
      "action": "encryptionReport",
      "encryptionReport": (
        "encryptionProfiles": [
          {
            "profile": "Alice's Profile"
          },
          {
            "profile": "Bob's Profile"
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionProfiles[ ].comment

Description A string. Information that may be helpful for the user.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "encryptionProfiles": [
          {
            "comment": "Use with bank forms",
            "profile": "Alice's"
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionProfiles[ ].profile

Description A string. Information that may be helpful for the user.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "encryptionProfiles": [
          {
            "profile": "Alice's"
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionProfiles[ ].vendor

Description A string. The UUID of the owner of the encryptionProfile. If the application recognizes the UUID then it may examine the rest of the object for content specific to that vendor.

Presence Optional.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "encryptionProfiles": [
          {
            "comment": "Acme",
            "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
            "profile": "Alice's",
            ...
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionPublicKeys

Description An array of zero or more encryptionPublicKey objects.

Encryption public keys are sent to the scanner prior to capturing images. They are not stored in the scanner.

Presence Optional. If the array is not included in the report, the scanner does not support public keys. If the array is present but empty, the scanner supports public keys. At this time any other content must be vendor specific.

There are no public keys when a session is started. All public keys are discarded when a task completes or times out.

Members comment
vendor

Examples

The scanner does not support public keys.

```
{
  "actions": [
    {
      "action": "encryptionReport"
      "encryptionReport": (
      }
    }
  ]
}
```

The scanner supports public keys.

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "encryptionPublicKeys": [
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionPublicKeys[ ].comment

Description A string. Information that may be helpful for the user.

Presence Optional.

Values Any valid UTF8-encoded JSON string.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "encryptionPublicKeys": [
          {
            "comment": "Acme",
            "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
            ...
          }
        ]
      }
    }
  ]
}
```

task.actions[ ].encryptionReport.encryptionPublicKeys[ ].vendor

Description A string. The UUID of the owner of the encryptionPublicKey. If the application recognizes the UUID then it may examine the rest of the object for content specific to that vendor.

Presence Optional.

Values A vendor unique UUID.

Examples

```
{
  "actions": [
    {
      "action": "encryptionReport",
      "encryptionReport": {
        "encryptionPublicKeys": [
          {
            "comment": "Acme",
            "vendor": "8f23d778-120e-11e5-9493-1697f925ec7b",
            ...
          }
        ]
      }
    }
  ]
}
```

TWAIN Direct Attributes

Overview

This section describes all of the attributes supported by TWAIN Direct. Each attribute may contain one of the following:

Description - an explanation of the attribute.

Scope - the portion of the scanner affected by the attribute such as:

- stream, this attribute only needs to appear once in a stream, if it appears more than once, then the first occurrence is used by the scanner (ex: alarms, doubleFeedDetection, etc).

- pixelFormat, this needs to be specified for each pixelFormat in a stream (ex: compression, resolution, etc)

Values - the allowed values for the attribute, one of which may be recommended as a default. This recommendation is a guide for scanner vendors. Application writers must not assume that it's the value they will get from any scanner.

See Also - items that are related to this attribute.

Examples - examples showing the use of the attribute.

alarms

Description Selects the scanner's audible alarms. Use the alarmVolume attribute to control the volume of the sounds.

Scope stream.

Values An array of strings specifying which alarms to use. Send an empty array to turn off all alarms. There is no recommended default.

all	When present, all alarms are turned on.
barcode	A barcode has been detected on the sheet of paper.
feederError	Any feeder errors that stops scanning, such as paper jams, or double feeders.
feederWarning	Any recoverable feeder warnings, such as feeder empty.
patchCode	A patch code has been detected on the sheet of paper.
power	Any power alerts, such as low battery, or if the scanner is going into a power conserving state.

See Also alarmVolume.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "alarms",
                      "values": [
                        {
```

```
                            "value": [ "all" ]
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
```

alarmVolume

Description Specifies the volume of the scanner's audible alarms.

Scope stream.

Values Number (positive integer).

1 - 100	1 is the lowest volume, and 100 is full volume. Use the alarms attribute to turn alarms on and off. There is no recommended default.

See Also alarms.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "alarmVolume",
                      "values": [
                        {
                          "value": 50
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

automaticDeskew

Description Corrects the skew of an image.

Scope pixelFormat.

Values String, one of the following.

on	Corrects image skew, rotating each image so that edges are close to square as possible. This rotation is not based on content, just the image's deviation from 0-degrees.
off	Returns images with no skew correction.

See Also flipRotation, mirror, rotation.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "automaticDeskew",
                      "values": [
                        {
                          "value": "on"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

automaticSize

Description Forces the output image to match either the dimensions specified by the sheetSize attribute, or the closest match in the list of allowed values defined by sheetSize.

Scope pixelFormat.

Values String, one of the following.

off	Make no changes to the size of the image. This is the recommended default.
automatic	Change the dimensions of the image to be the same as the closest match listed under the sheetSize attribute. This works best when cropping is set to automatic and automaticDeskew is on.
sheetSize	Force the dimensions of the image to match the current value of sheetSize.

See Also automaticDeskew, cropping, sheetSize.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "automaticSize",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
```

```
            ]
          }
        ]
      }
    ]
}
```

barcodes

Description Set this attribute to detect barcodes on the scanned images. Barcode data is returned in the image's metadata.

Scope pixelFormat.

Values An array of strings specifying which barcodes to detect. Send an empty array to turn off barcode detection. There is no recommended default.

all	All barcodes recognized by this scanner.
2Of5DataLogic	Variant of 2 of 5 barcodes using both black and white bars.
2Of5Iata	Variant of 2 of 5, used by the airline industry.
2Of5Industrial	Variant of 2 of 5, used by photofinishing and warehouse sorting.
2Of5Interleaved	Variant of 2 of 5, which encodes pairs of numbers.
2Of5Matrix	Variant of 2 of 5.
2Of5NonInterleaved	Variant of 2 of 5, has no check digits.
3Of9	Also called Code 39, limited to 43 characters.
3Of9FullAscii	Variant of 3 of 9, using pairs of characters to represent ASCII.
codabar	Also called Code 2 of 7, used in libraries.
codabarWithStartStop	Variant of codabar.
code128	Codes first 128 of ASCII (binary data).
code93	High density and security enhancements to 3Of9.
ean13	Variant of UPC, used in retail.
ean8	Smaller variant of EAN-13.

maxicode	Used to track and manage packages.
pdf417	Used in transport, ID cards and inventory management.
postnet	Used to assist in directing mail (ZIP Codes).
qrCode	2D code, popular with phones.
ucc128	Also called GS1-128, used in retail.
upca	UPC variant, used in retail.
upce	UPC variant, used for smaller packages in retail.

See Also micr, patchCodes.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "barcodes",
                      "values": [
                        {
                          "value": [ "bc3Of9", "qrcode" ]
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

bitDepthReduction

Description Tells the scanner the halftoning algorithm it should use to reduce an image's bit depth from grayscale (gray8), when creating a black-and-white (bw1) image. This is done to improve compression, and may be used to improve the readability of text.

Scope pixelFormat.

Values String, one of the following.

errorDiffusion	Use this to enhance text.
dynamic	The scanner automatically determines the best halftoning for the image. This is the recommended default.
thresholding	This is the simplest halftoning technique. Pixel values below the threshold attribute value are set to black. Pixel values at or above the threshold are set to white.

See Also pixelFormat, threshold.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "bitDepthReduction",
                      "values": [
                        {
                          "value": "dynamic"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
```

```
            }
          ]
        }
    ]
}
```

brightness

Description Select the amount of white appearing in the image.

Scope pixelFormat.

Values Number (positive integer), or one of the string listed below.

0 - 100	The brightness value, with 0 providing the darkest setting and 100 providing the brightest setting. The recommended default is 50.
automatic	Automatically select the best brightness for the image.

See Also contrast.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "brightness",
                      "values": [
                        {
                          "value": 50
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

compression

Description Indicates the kind of compression that the scanner applies to the image. Use compression to reduce the number of bytes needed for the images coming out of the scanner, this improves communication performance and results in smaller image files being stored on disk.

Scope pixelFormat.

Values String, one of the following.

autoVersion1	Automatically selects group4 or jpeg compression based on the pixelFormat. This is the recommended default.
group4	CCITT Group 4 Fax, appears in the PDF/raster as /Filter /CCITTFaxDecode (with /K -1). For pixelFormat bw1 only.
jpeg	JPEG Baseline, appears in the PDF/Raster as /Filter /DCT. For pixelFormat rgb24 and gray8 only.
none	Uncompressed rasters, with each raster line aligned on a 4-byte boundary. Appears in the PDF/raster as /Filter null.

See Also pixelFormat.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "compression",
                      "values": [
                        {
                          "value": "jpeg"
                        }
                      ]
```

```
                        }
                    ]
                }
            ]
        }
    ]
}
```

continuousScan

Description Controls the method a scanner's automatic document feeder uses to feed each sheet of paper.

Scope stream.

Values String, one of the following.

off	The scanner scans one sheet of paper at a time. The next sheet is not fed into the scanner until all of the images from the previous sheet have been released by the application. Use this setting to allow each sheet to be examined before allowed the next sheet to be scanned.
on	The scanner is free to take in sheets of paper as fast as it can. Use this setting for the best performance. This is the recommended default.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "continuousScan",
                      "values": [
                        {
                          "value": "on"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

contrast

Description Selects how soft or crisp the image looks.

Scope pixelFormat.

Values Number (positive integer), or one of the string listed below.

0 - 100	The contrast value, with 0 providing the least contrast (images that may look muddy), and 100 providing the most contrast (images that consist of white and black or colored regions). The recommended default is 50.
automatic	Automatically select the best contrast for the image.

See Also brightness.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "contrast",
                      "values": [
                        {
                          "value": 50
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

cropping

Description Select the method used by the scanner to locate the region of interest, that is, the portion of the sheet of paper that the user wants to capture.

Scope pixelFormat.

Values String, one of the following.

automatic	Automatically detect the cropping area for the full sheet of paper. If scanning from a flatbed the image will include all of the items found on the glass surface. This is the recommended default.
automaticMultiple	Automatically detect the cropping area for the full sheet of paper for an automatic document feeder. If scanning from a flatbed the scanner will look for multiple images (such as photographs or business cards) and will return one image for each item that it finds.
fixed	Crop the image using the region of interest specified by the height/width/offsetX/offsetY attributes.
fixedAutomaticLength	Crop the image using the height/width/offsetX/offsetY attributes. The scanner will automatically detect the length of the sheet, and so may return an image shorter than that specified by the fixed region of interest.
long	Used for long documents. Divide the image into multiple pieces using the height/width/offsetX/offsetY attributes. Each portion appears as its own /XObject in the finished PDF/raster. The offsetx and offsety values are measured from the left side of the transport or camera.
relative	Find the borders of the sheet of paper, and then select a region within that using the height/width/offsetX/offsetY attributes, where the offsetx and offsety are located in the upper left corner of the scanned side of the sheet.

See Also height, offsetX, offsetY, overScan, sheetSize, width.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "cropping",
                      "values": [
                        {
                          "value": "auto"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

discardBlankImages

Description When turned on the scanner discards images that are considered to be blank. Use this to improve scanning performance, since it's not necessary to transfer blank images from the scanner. The image metadata can be used to detect when an image or even an entire sheet has been discarded.

Scope pixelFormat.

Values String, one of the following.

off	Do not discard blank images. This is the recommended default.
on	Discard images that the scanner considers blank.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "discardBlankImages",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

doubleFeedDetection

Description Enable or disable the ability of the scanner to detect when more than one sheet of paper is fed into the scanner. Examples include when sheets are stapled together or just stuck together.

Scope stream.

Values String, one of the following. There is no recommended default.

off	Do not watch for double feeds.
on	Detect double feeds.

See Also doubleFeedDetectionLength, doubleFeedDetectionResponse, doubleFeedDetectionSensitivity.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "doubleFeedDetection",
                      "values": [
                        {
                          "value": "on"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

doubleFeedDetectionLength

Description Detects double feeds based on the length of the physical sheets of paper being fed into the scanner. For instance, if the user indicates that they are scanning A4 sheets of paper and the scanner sees a sheet that appears to be 14” (35.56cm) in length, it assumes that two sheets of paper are stuck together and treats that as a double feed.

This attribute is ignored if doubleFeedDetection is set to “off”.

Scope stream.

Values Number (positive integer in microns). There is no recommended default.

0 - n	If the value is 0, then double feeds are not detected by length. For values great this 0 the action specified by the doubleFeedDetectionResponse is applied if the length of the sheet of paper exceeds this number. Scanner manufacturers may silently add a small value to this number to account for document skew.
closest	Selects the supported value closest to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If the previous value requested is exactly between two choices, then the higher value is selected (ex: 100 and 200 are valid, and 150 is requested, then 200 is used). If the previous value is entirely out of range, the scanner’s minimum or maximum value is used, whichever is closer.
closestGreaterThan	Selects the closest supported value greater than or equal to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If there is no greater value, the scanner’s maximum value is used.
closestLessThan	Selects the closest supported value less than or equal to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If there is

	no lesser value, the scanner's minimum value is used.
maximum	Use the maximum value supported by this scanner.
minimum	Use the minimum value supported by this scanner.

See Also doubleFeedDetection, doubleFeedDetectionResponse, doubleFeedDetectionSensitivity.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "doubleFeedDetectionLength",
                      "values": [
                        {
                          "value": 279400
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

doubleFeedDetectionResponse

Description Determines the response to take if a double feed is detected by the scanner. This attribute is ignored if doubleFeedDetection is set to “off”.

Scope stream.

Values Array, zero or more of the following. There is no recommended default.

alarm	Generate an audible alarm when a double feed is detected.
stop	Stop capturing images with an error.
pauseAndWait	Pause the capturing of images. The user is presented with options on the scanner.
doNotPrint	Do not print on sheets of paper that trigger a double feed.

See Also alarms, doubleFeedDetection, doubleFeedDetectionLength, doubleFeedDetectionSensitivity.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "doubleFeedDetectionResponse",
                      "values": [
                        {
                          "value": [ "alarm", "donotprint" ]
                        }
                      ]
                    }
                  ]
                }
              ]
            }
```

```
            ]
          }
        ]
      }
    ]
}
```

doubleFeedDetectionSensitivity

Description Determines the sensitivity of the double feed detection system. This is intended for scanners that use sensors to detect the presence of two or more sheets of paper entering the scanner's feeder. This attribute is ignored if doubleFeedDetection is set to "off".

Scope stream.

Values String, one of the following.

high	Aggressively detect any possible double feeds. This is the best setting if all the sheets of paper are from the same stock with no labels attached.
low	Use this setting to ignore labels, and thick or wrinkled sheets of paper. This is the recommended default.
medium	Use this when scanning good quality sheets of paper from varying stocks. It should still ignore most labels attached to the sheet of paper.

See Also doubleFeedDetection, doubleFeedDetectionLength, doubleFeedDetectionResponse.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "doubleFeedDetectionSensitivity",
                      "values": [
                        {
                          "value": [ "low" ]
                        }
                      ]
                    }
```

```
                    ]
                }
            ]
        }
    ]
}
```

flipRotation

Description This attribute describes the orientation of the front image to the rear image on each sheet of paper, so that the scanner can correctly rotate the rear image.

Scope pixelFormat.

Values String, one of the following.

book	The right edge of the front image is connected to the left edge of the rear image, and the pattern repeats from there. Put another way, after viewing the front image one switches to the rear image by rotating the sheet of paper 180 degrees around the vertical axis. This is the recommended default, and is normal for most documents.
fanFold	The bottom edge of the front image is connected to the top edge of the rear image, and the pattern repeats from there. Put another way, after viewing the front image one switches to the rear image by rotating the sheet of paper 180 degrees around the horizontal axis.

See Also automaticDeskew, mirror, rotation.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "flipRotation",
                      "values": [
                        {
                          "value": "book"
                        }
                      ]
                    }
```

```
                    ]
                }
            ]
        }
    ]
}
```

height

Description The height of the captured image in microns.

Scope stream.

Values Number (positive integer), or one of the string listed below. There is no recommended default.

1 - n	An integer value in microns. The conversion of inches to microns is (inches * 25400), so 11 inches is 279400 microns. If the size of the value exceeds 2147483647 (2^{32}-1), then the value must be sent as a string.
maximum	Use the maximum height supported by the scanner, adjusted for the selected offsetY value. This can be used in concert with a value of “minimum” for the offsetY.
minimum	Use the minimum height supported by the scanner. This can be used in concert with a value of “maximum” for the offsetY.

See Also cropping, offsetX, offsetY, sheetSize, width.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "height",
                      "values": [
                        {
                          "value": 279400
                        }
                      ]
                    }
                  ]
                }
```

```
                ]
              }
            ]
          }
        ]
      }
    ]
}
```

imageMerge

Description Merges the front and rear image of a document in one of four orientations: front above the rear, front below the rear, front to the left of the rear, or front to the right of the rear.

This attribute only has meaning when the scanner is asked to capture both the front and rear of the sheet of paper.

The size of the final image is two times the size of the image that has the largest width and the image that has the largest height. For instance, if the dimension of the front image is 100 x 200 and the dimension of the rear image is 200 x 100, then the final image will be 400 x 400.

In the merged image each side (front and rear) has its origin in the upper left hand corner of its segment.

It is recommended that this attribute is used with a source of feeder. If used with feederFront and feederRear, the scanner is free to make its own decisions on how to merge differing properties, such as pixelFormat or compression.

Scope stream.

Values String, one of the following.

off	Do not merge the front and rear images. This is the recommended default.
frontAboveRear	In the finished image the front of the sheet is shown above the rear of the sheet.
frontBelowRear	In the finished image the front of the sheet is shown below the rear of the sheet.
frontLeftOfRear	In the finished image the front of the sheet is shown to the left of the rear of the sheet.
frontRightOfRear	In the finished image the front of the sheet is shown to the right of the rear of the sheet.

See Also imageMergeHeightThreshold, source.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "imageMerge",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

imageMergeHeightThreshold

Description Selects the height threshold for merging images, when both a front and a rear image are present.

If the height of both the front and rear images are less than or equal to this value, they are merged according to the setting of imageMerge.

If the height of either the front or the rear images are more than this value, they are not merged.

Scope stream.

Values Number (positive integer in microns).

0 - n	A value of 0 indicates that all front and rear images must always be merged.

See Also imageMerge.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "imageMergeHeightThreshold",
                      "values": [
                        {
                          "value": 1
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
```

```
            }
         ]
      }
   ]
}
```

invert

Description Inverts each pixel in the image. For bw1 this means 1's becomes 0's and 0's become 1's. For gray8 and rgb24, each channel is flipped using the equation (255 - current value). This has the effect of creating a negative image from the original sheet of paper.

Scope pixelFormat.

Values String, one of the following.

off	Do not invert the image. This is the recommended default.
on	Invert the pixel values of the image.

See Also pixelFormat.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "invert",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

jpegQuality

Description Selects the quality of a JPEG image. JPEG is a lossy compression, so there is a trade-off between the quality of a compressed image and its size. Images that are more highly compressed have lower quality, meaning loss of detail and creation of artifacts.

This attribute only has meaning if the the compression attribute results in the scanner creating a JPEG compressed image.

Scope pixelFormat.

Values Number (positive integer), or one of the strings listed below.

1 - 100	Lower values result in a lower image quality and a smaller image size. Higher values result in a higher image quality and a larger image size. Scanner vendors translate this number to represent the widest range of JPEG quantization tables supported by their hardware. Applications writers are warned that low values may result in images with extremely low quality.
closest	Selects the supported value closest to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If the previous value requested is exactly between two choices, then the higher value is selected (ex: 10 and 20 are valid, and 15 is requested, then 20 is used). If the previous value is entirely out of range, the scanner's minimum or maximum value is used, whichever is closer.
closestGreaterThan	Selects the closest supported value greater than or equal to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If there is no greater value, the scanner's maximum value is used.
closestLessThan	Selects the closest supported value less than or equal to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If there is no lesser value, the scanner's minimum value is used.

best	Improves image quality to its highest recommended setting, but still allows for some meaningful amount of compression. Best has higher image quality than better, but not as good as maximum. The scanner responds with best, not a number.
better	Improves image quality at the expense of a larger image size. Better has better image quality than good, but not as good as best. The scanner responds with better, not a number.
good	Represents a compromise between the size and the quality of the image, and is suitable in most situations. Good has better image quality than minimum, but not as good a better. This is the recommended default. The scanner responds with good, not a number.
maximum	Maximizes the image quality, file sizes may be very large (but still smaller than images with no compression). The scanner responds with maximum, not a number.
minimum	Minimizes the image quality, this creates the smallest image size recommended as being useful by the scanner vendor. The scanner responds with minimum, not a number.

See Also compression.

Examples

```
{
  "comment": "Set jpegQuality to good.",
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "jpegQuality",
                      "values": [
                        {
                          "value": "good"
                        }
                      ]
                    }
```

```
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
}
```

micr

Description Magnetic Ink Character Recognition. Use this to enable or disable the scanner's ability to recognize MICR data, which is used to validate the legitimacy of documents, most often for checks.

Scope pixelFormat.

Values String, one of the following.

off	Do not look for MICR data. This is the recommended default.
on	Detect MICR data, and when found return it in the metadata.

See Also barcodes, patchCodes.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "micr",
                      "values": [
                        {
                          "value": "on"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

mirror

Description Flips the image around its horizontal or vertical access, resulting in a mirror image of the original sheet.

Scope pixelFormat.

Values String, one of the following.

off	Do not mirror the image. This is the recommended default.
horizontal	Flip the image around its horizontal axis.
vertical	Flip the image around its vertical axis.
verticalAndHorizontal	Flip the image around its vertical and horizontal axis.

See Also automaticDeskew. flipRotation, rotation.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "mirror",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

noiseFilter

Description Removes noise (small isolated pixels) from an image. Do this to improve compression.

Scope pixelFormat.

Values String, one of the following.

off	Turns off noise filtering. This is the recommended default.
automatic	The scanner selects the best noise filter.
lonePixel	Only modify pixels that are completely surrounded by pixels of the same color.
majorityRule	Modify pixels that are mostly surrounded by pixels of the same color..

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "noiseFilter",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

overScan

Description .

Scope pixelFormat.

Values String, one of the following.

off	Turns off overscan. For fixed cropping the boundaries will match the region of interest specified by the application. This is the recommended default.
on	Turns on overscan. The scanner adds data beyond the fixed cropping region of interest to help an application support its own automatic crop and deskew algorithm. This value only has meaning if cropping is set to “fixed”.

See Also cropping.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "overScan",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
```

```
  ]
}
```

numberOfSheets

Description The number of sheets that the scanner will capture. If the scanner runs out of sheets before reaching this number scanning will conclude, it won't wait for more paper.

Scope stream.

Values Number (positive integer), or one of the strings listed below.

1 - n	The number of sheets.
maximum	Capture the maximum number of sheets supported by the scanner. This is the recommended default.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "numberOfSheets",
                      "values": [
                        {
                          "value": "maximum"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
```

```
            }
          ]
        }
    ]
}
```

offsetX

Description The horizontal offset of the captured image in microns.

Scope pixelFormat.

Values Number (positive integer), or one of the string listed below. There is no recommended default.

1 - n	An integer value in microns. The conversion of inches to microns is (inches * 25400). If the size of the value exceeds 2147483647, then the value must be sent as a string.
maximum	Use the maximum horizontal offset supported by the scanner, adjusted for the selected width value if offsetx and width are in conflict. This can be used with a value of “minimum” for width.
minimum	Use the minimum horizontal offset supported by the scanner. This can be used with a value of “maximum” for width.

See Also cropping, height, offsetY, sheetSize, width.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "offsetX",
                      "values": [
                        {
                          "value": 25400
                        }
                      ]
                    }
                  ]
                }
```

```
                ]
              }
            ]
          }
        ]
      }
    ]
}
```

offsetY

Description The vertical offset of the captured image in microns.

Scope pixelFormat.

Values Number (positive integer), or one of the string listed below. There is no recommended default.

1 - n	An integer value in microns. The conversion of inches to microns is (inches * 25400). If the size of the value exceeds 2147483647, then the value must be sent as a string.
maximum	Use the maximum vertical supported by the scanner, adjusted for the selected height value if offsety and height are in conflict. This can be used with a value of “minimum” for height.
minimum	Use the minimum vertical offset supported by the scanner. This can be used with a value of “maximum” for height.

See Also cropping, height, offsetX, sheetSize, width.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "offsetY",
                      "values": [
                        {
                          "value": 25400
                        }
                      ]
                    }
                  ]
                }
```

```
                    ]
                }
            ]
        }
    ]
  }
}
```

patchCodes

Description Set this attribute to detect patch codes on the scanned images. Patch code data is returned in the image's metadata, and depending on the scanner, may make internal changes in the scanner, as well.

The image level statements only have meaning if the scanner supports image addressing.

Scope pixelFormat.

Values An array of strings specifying which patch codes to detect. Send an empty array to turn off patch code detection. An empty array is the recommended default.

all	Detect all of the patch codes supported by the scanner.
patch1	Patch Code 1.
patch2	Patch Code 2, assigns image level 2 to the current document.
patch3	Patch Code 3, assigns image level 3 to the current document.
patch4	Patch Code 4, feature patch, causes the scanner to take some action.
patchT	Transfer Patch, assigns a predefined image level (usually 2 or 3) to the next document (transfer patch).
patch6	Patch Code 6.
patch7	Patch Code 7.
patch8	Patch Code 8.
patch9	Patch Code 9.
patch10	Patch Code 10.
patch11	Patch Code 11.

patch12	Patch Code 12.
patch13	Patch Code 13.
patch14	Patch Code 14.
patch15	Patch Code 15.

See Also barcodes, micr.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "patchCodes",
                      "values": [
                        {
                          "value": [ "patch2", "patch3" ]
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

resolution

Description The resolution of the image in dots-per-inch (dpi). Small values capture smaller image sizes with less detail. Large values result in larger image sizes with more detail.

Scope pixelFormat.

Values Number (positive integer), or one of the strings listed below. There is no recommended default.

1 - n	An integer value. Typical values are 75, 100, 150, 200, 240, 250, 300, 400, 500, 600, 1200, 2400, 4800, 9600 and 19200. Some scanners support continuous ranges.
closest	Selects the supported value closest to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If the previous value requested is exactly between two choices, then the higher value is selected (ex: 100 and 200 are valid, and 150 is requested, then 200 is used). If the previous value is entirely out of range, the scanner's minimum or maximum value is used, whichever is closer.
closestGreaterThan	Selects the closest supported value greater than or equal to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If there is no greater value, the scanner's maximum value is used.
closestLessThan	Selects the closest supported value less than or equal to the previously requested value. If used alone, or if the previous value is not an integer, the scanner uses its power-on default. If there is no lesser value, the scanner's minimum value is used.
maximum	Use the maximum value supported by this scanner.
minimum	Use the minimum value supported by this scanner.
optical	Use the optical resolution of the scanner. This produces the best image (least artifacts from scaling) that the scanner can produce,

but the images can be large if the optical resolution is high, such as 1200dpi, which will result in a 404MB uncompressed color image for an 8.5 x 11 inch sheet of paper. If a scanner does not have an optical value it uses its power-on default.

preview Use the minimum resolution of the scanner suitable for creating preview images.

Examples

In this example the scanner is asked for 280, and if that's not available, the closest valid value greater than the one requested is used, which might be 300.

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "resolution",
                      "values": [
                        {
                          "value": 280
                        },
                        {
                          "value": "closestGreaterThan"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

rotation

Description Rotate the image captured by the scanner.

Scope pixelFormat.

Values Number (positive integer), or one of the string listed below.

0	Rotate the image 0 degrees (or in other words, leave it alone).
90	Rotate the image 90 degrees in a clockwise direction.
180	Rotate the image 180 degrees in a clockwise direction.
270	Rotate the image 270 degrees in a clockwise direction.
automatic	Automatically rotate the image based on data the scanner finds in it, such as text. This is the recommended default.

See Also automaticDeskew, flipRotation, mirror.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "rotation",
                      "values": [
                        {
                          "value": "automatic"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
```

```
            }
          ]
        }
    ]
}
```

sheetHandling

Description Tells the scanner the kind of paper that will be passing through its automatic document feeder. Use this to reduce paper jams, or to tell the scanner to use extra care to avoid damaging items. Note that some of the settings will likely reduce the speed of scanner to protect the paper passing through it.

Scope stream.

Values String, one of the following.

fragile	Paper that is thin or easily torn.
normal	Normal paper, as used with most printers. This is the recommended default, and allows the scanner to run at its full speed.
photograph	Photographic paper. Use this to reduce the risk of damage.
thick	Paper that is thicker than standard stock.
trifold	Trifold paper, as in a brochure or some maps. This setting may also be useful for paper that's curled or wrinkled.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "sheetHandling",
                      "values": [
                        {
                          "value": "normal"
                        }
                      ]
                    }
```

```
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
}
```

sheetSize

Description Sets the height, offsetX, offsetY and width values based on the selected item. The list is a mixture of ISO and English measures. The ISO items are used as follows:

a0,a1	Technical drawings, posters
a2,a3	Drawings, diagrams, large tables
a4	Letters, magazines, forms, catalogs, etc
a5	Note pads
a6	Postcards
b5,a5,b6,a6	Books
c4,c5,c6	Envelopes for a4 letters: unfolded (c4), folded once (c5), folded twice (c6)
b4,a3	Newspapers

Scope pixelFormat.

Values String, one of the following. There is no recommended default.

din4A0	1682000µm x 2378000µm.
din2A0	1189000µm x 1682000µm.
isoA0	841000µm x 1189000µm.
isoA1	594000µm x 841000µm.
isoA2	420000µm x 594000µm.
isoA3	297000µm x 420000µm.
isoA4	210000µm x 297000µm.
isoA5	148000µm x 210000µm.
isoA6	105000µm x 148000µm.
isoA7	74000µm x 105000µm.

isoA8	52000µm x 74000µm.
isoA9	37000µm x 52000µm.
isoA10	26000µm x 37000µm.
isoB0	1000000µm x 1414000µm.
isoB1	707000µm x 1000000µm.
isoB2	500000µm x 707000µm.
isoB3	353000µm x 500000µm.
isoB4	250000µm x 353000µm.
isoB5	176000µm x 250000µm.
isoB6	125000µm x 176000µm.
isoB7	88000µm x 125000µm.
isoB8	62000µm x 88000µm.
isoB9	44000µm x 62000µm.
isoB10	31000µm x 44000µm.
isoC0	917000µm x 1297000µm.
isoC1	648000µm x 917000µm.
isoC2	458000µm x 648000µm.
isoC3	324000µm x 458000µm.
isoC4	229000µm x 324000µm.
isoC5	162000µm x 229000µm.
isoC6	114000µm x 162000µm.

isoC7	81000µm x 114000µm.	
isoC8	57000µm x 81000µm.	
isoC9	40000µm x 57000µm.	
isoC10	28000µm x 40000µm.	
jisB0	1030000µm x 1456000µm.	
jisB1	728000µm x 1030000µm.	
jisB2	515000µm x 728000µm.	
jisB3	364000µm x 515000µm.	
jisB4	257000µm x 364000µm.	
jisB5	182000µm x 257000µm.	
jisB6	128000µm x 182000µm.	
jisB7	91000µm x 128000µm.	
jisB8	64000µm x 91000µm.	
jisB9	45000µm x 64000µm.	
jisB10	32000µm x 45000µm.	
usBusinessCard	88900µm x 50800µm	(3.5” x 2.0”).
usExecutive	184150µm x 266700µm	(7.25” x 10.5”).
usLedger	279400µm x 431800µm	(11.0” x 17.0”).
usLegal	215900µm x 355600µm	(8.5” x 14.0”).
usLetter	215900µm x 279400µm	(8.5” x 11.0”).
usStatement	139700µm x 215900µm	(5.5” x 8.5”).

See Also cropping, height, offsetX, offsetY, width.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "sheetSize",
                      "values": [
                        {
                          "value": "isoA4"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

threshold

Description When bitDepthReduction is set to thresholding, this value selects the cut off point for deciding which grayscale pixels are set to black and which are set to white in the finished black-and-white (bw1) image.

Scope pixelFormat.

Values Number (positive integer).

0 - 255	Selects the threshold value. Grayscale pixel values less than this value are set to black. Grayscale pixel values equal to or greater than this value are set to white. The recommended default is 128.

See Also bitDepthReduction.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "threshold",
                      "values": [
                        {
                          "value": 128
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

uncalibratedImage

Description Allows the application to get uncalibrated image data from the scanner. This maximizes the dynamic range of the image data for applications that need to do their own image processing. Applications that do this need to avoid lossy compressions, like jpeg.

Scope pixelFormat.

Values String, one of the following.

off	Calibrate the data so the scanner can produce the best quality images. This is the recommended default.
on	Do not calibrate the image data. This is sometimes referred to as raw output, but it's worth noting that some processing is still taking place before the image gets to the application.

See Also compression.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "uncalibratedImage",
                      "values": [
                        {
                          "value": "off"
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
```

```
      }
   ]
}
```

width

Description The width of the captured image in microns.

Scope pixelFormat.

Values Number (positive integer), or one of the string listed below.

1 - n	An integer value in microns. The conversion of inches to microns is (inches * 25400), so 8.5 inches is 215900 microns. If the size of the value exceeds 2147483647 (2^{32}-1), then the value must be sent as a string.
maximum	Use the maximum width supported by the scanner, adjusted for the selected offsetX value. This can be used in concert with a value of “minimum” for offsetX.
minimum	Use the minimum width supported by the scanner. This can be used in concert with a value of “maximum” for offsetX.

See Also cropping, height, offsetX, offsetY, sheetSize.

Examples

```
{
  "actions": [
    {
      "action": "configure",
      "streams": [
        {
          "sources": [
            {
              "pixelFormats": [
                {
                  "attributes": [
                    {
                      "attribute": "width",
                      "values": [
                        {
                          "value": 215900
                        }
                      ]
                    }
                  ]
                }
              ]
```

```
                }
            ]
        }
    ]
  }
]
}
```