Mopria Alliance Technical Working Group

eSCL Technical Specification

Version 2.97

January 17, 2025

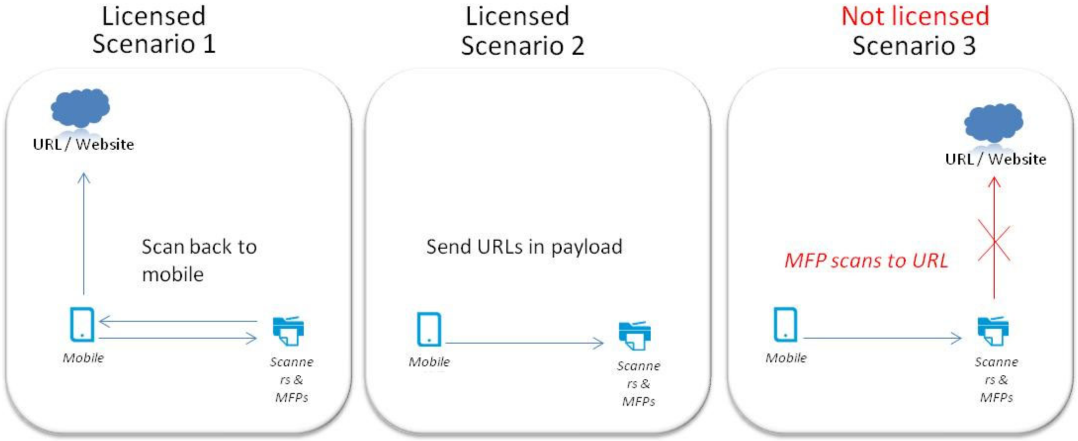
Approved by the Mopria Board

MOPRIA ALLIANCE PROPRIETARY - SUBJECT TO CHANGE WITHOUT NOTICE
This document may be used with the permission of Mopria Alliance under the terms set forth herein. Use of this document by anyone who is not a member of Mopria Alliance (each such person, a "Member") is prohibited.
The use of this document is governed by the terms of the Mopria IPR Policy and the terms set forth herein and by your use of the document, you are agreeing to these additional terms set forth herein. Unless this document is clearly designated as an approved specification, this document is a work-in-process and is not an approved Mopria Alliance specification. This document is subject to revision or removal at any time without notice. Information contained in this document may be used at your sole risk. Mopria Alliance assumes no responsibility for errors or omissions in this document. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF ANY THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS, AND COMPLIANCE WITH APPLICABLE LAWS. This copyright permission does not constitute an endorsement of any implementations or any other products or services. Mopria Alliance trademarks and certification marks may not be used unless specifically licensed by or without the written permission of Mopria Alliance.
Mopria Alliance has not conducted an independent intellectual property rights ("IPR") review of this document and the information contained herein. Implementations based upon this document may infringe upon patents of Members or third parties. This document is a Final Specification of Mopria Alliance. The IPR Policy of Mopria Alliance provides for the royalty-free patent licensing commitment of Members for Final Specifications developed in Working Groups that have selected the "Royalty Free License" scheme. This Final Specification has been developed in a Working Group that has selected the "Royalty Free License" scheme. The details of such licensing commitments are contained in the IPR Policy of Mopria Alliance and Members should review such IPR Policy before making, using or selling implementations based upon this document. Mopria and each Member, including Working Group representatives and Working Group Chairs, take no position as to whether any intellectual property rights exist in any Draft or Final Specifications.
Mopria Alliance owns the copyright in this document and reserves all rights therein. A user of this document may duplicate and distribute copies of the document in connection with the authorized uses described herein, provided any duplication in whole or inpart includes the copyright notice and the disclaimer text set forth herein. Unless prior written permission has been received from Mopria Alliance, any other use of this document and all other duplication and distribution of this document are prohibited.
Unauthorized use, duplication, or distribution is an infringement of Mopria Alliance's copyright; and any such unauthorized use, duplication, or distribution may result in a Member's membership being terminated.
Each Member is solely responsible for the compliance of their products with applicable laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their products related to such regulations within the applicable jurisdictions.

NOTHING IN THIS DOCUMENT CREATES ANY WARRANTIES (WHETHER EXPRESS OR IMPLIED) REGARDING SUCH LAWS OR REGULATIONS. MOPRIA ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS DOCUMENT AND ANY INFORMATION CONTAINED IN THIS DOCUMENT. ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS RELATING TO USE OF THE DOCUMENT IS EXPRESSLY DISCLAIMED. BY USE OF THE DOCUMENT, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST MOPRIA ALLIANCE AND ITS MEMBERS RELATED TO USE OF THE DOCUMENT. “Notwithstanding the Copyright License Grant in Part 1 (b) of Attachment A “License to Final Specification” of the Intellectual Property Rights Policy of the Mopria Alliance, Inc. Membership Agreement, the copyright license grant to HP to the Final Specification based on the eSCL Contribution from HP shall include the right to modify, create derivative works, distribute and sublicense the Final Specification (including any Contribution of any Member to the Final Specification) to third parties.

By contributing the eSCL specification to the Mopria Alliance, HP allows claims in US7263524 and no other HP Intellectual property is allowed or implied. See the following page for specific uses cases:

Claims in US7263524 allow for Scenarios 1 & 2 and do not allow Scenario 3.



Document History

Version	Date	Status	Comments
2.9	2021-03-22	Final	Approved by the Mopria Board
2.91	06/08/2022		Added OCR setting
2.92	09/30/2022		Added OCR language support
2.93	11/10/2022		Added 2.91 and 2.92 OCR features to the 2.9 final specification
2.94	04/10/2023	Draft	Updated errata for schema location ActualWidth and ActualHeight attributes
2.95	08/10/2023		Added support for default attribute to DiscreteResolutions Added description for AdfState enums
2.96	05/13/2024		Added support for PDF encryption for PDF scans
2.97	08/12/2024		Updated Security section to include user management behaviour

Contents

1 Introduction

The eSCL specification defines the interfaces, data types and overall behavioral model for driving a scanner engine from various classes of clients: SW, Cloud Services, Mobile device, Embedded Web Server (EWS).

eSCL is a RESTful interface. Some of the XML element types are leveraged from the PWG Common definitions (namespace=http://www.pwg.org/schemas/2010/12/sm version 1.120). The remaining XML elements are defined in the eSCL schema (identified by the namespace http://schemas.hp.com/imaging/escl/2011/05/03).

One of the design principles of eSCL is to shift the focus — from a protocol where all the controls are given to the client application — to a model where the client does not need extensive knowledge of the scanner inner workings, and the client can be more focused on the user experience.

The concept of “Intent” is at the core of this principle.

1.1 Intents

Modern scanners are complex: many settings must be provided in order to control the scan engine accurately. Moreover, different scanner models can have very different behaviors and specifications. It makes it difficult for scan clients to understand what settings are required, what values are valid and authorized.

On the contrary, people’s interactions with a scanner mostly fit into a few use cases. Identifying, categorizing and modeling how people interact with a scanner result in a simpler and more intuitive client application design.

By focusing on people first, the Scan Intent simplifies the task for the scan client. Instead of specifying the settings expected by the scanner, the client declares what the 'intent' of the scan is. The scanner then finds the right combination of settings to better match the client's (and user’s) intention.

1.2 References

References provided below are for released publicly available documents.

[1] IEEE-ISTO PWG, http://www.pwg.org

[2] PWG Media Standardized Names 2.1 (PWG 5101.1) https://ftp.pwg.org/pub/pwg/candidates/cs-pwgmsn21-20230915-5101.1.pdf

[3] Bonjour Printing Specification 1.2 https://developer.apple.com/bonjour/printing-specification/

[4] RFC 6762 Multicast DNS http://tools.ietf.org/html/rfc6762

[5] RFC 6763 DNS-Based Service Discovery http://tools.ietf.org/html/rfc6763

[6] RFC 2782 A DNS Resource Record (RR) for specifying the location of services (DNS SRV) http://www.ietf.org/rfc/rfc2782.txt

[7] RFC 2817 “Upgrading to TLS Within HTTP/1.1” https://www.rfc-editor.org/rfc/rfc2817

[8] RFC 8446 “The Transport Layer Security (TLS) Protocol Version 1.3” https://www.rfc-editor.org/rfc/rfc8446

1.3 Definitions

The following definitions and terms are used in this document:

Print Client – Or just Client, is a device which renders content to transmit to the Printer.

Mopria Device – Or just Device, includes both Print Clients and Printers.

Plugin – Transforms the data from the Print sub-system to the PDL supported by the Printer.

1.4 Abbreviations and acronyms

IPP Internet Printing Protocol

PCLm Page Command Language for Mobile is the Page Description Language defined in the **Wi-Fi Direct Services Print Technical Specification** which is used to print rasterized pages.

PWG Raster Format (PWG 5102.4-2012)

PDL Page Description Language

PWG Printer Working Group

PDF Portable Document Format

WFA Wi-Fi Alliance™

2 HTTP Error Codes

Error Codes	Definitions
200 OK	The request was understood, and the scanner started the transaction
201 Created	A scan job is created
301 Moved	The client is redirected to a secure connection
400 Bad Request	Request could not be understood due to wrong syntax
401 Unauthorized	The client is challenged for access credentials
404 Not Found	Resource is not found

405 Method Not Allowed	Not a valid method for the URI requested.
409 Conflict	The parameters or payload are of the wrong types or values. Request not understood.
410 Gone	The resource (likely a job instance) used to exist but is gone.
416 Requested Range Not Satisfiable	The Range header in the request does not match the range of data the scanner has cached.
426 Upgrade Required	The scanner indicating that a client request cannot be completed without TLS [7].
500 Internal Server Error	The scanner could not perform the request for lack of resource or unknown internal error
503 Service Unavailable	The scanner isn’t momentarily able to attend this request. The client app should retry later.

3 Discovery

Network Scanner Discovery (Bonjour)

The scanner discovery is done using Bonjour: Multicast DNS (mDNS) [4] and DNS Service Discovery (DNS-SD) [5].

Required Service Type

Scanners conforming to the eSCL specification MUST register the service type: _uscan._tcp

Required Separate Registrations

Scanners conforming to this specification MUST register separate service names for logical devices rather than returning multiple TXT records for a single service name as described in Section 9.6 of the Bonjour Specification.

Scanner Advertisement

The scanner MUST advertise a SRV and a TXT record with a name in the form of: <Scanner Name>._uscan._tcp.local.

For example: HP Officejet Pro 8500._uscan._tcp.local.

The scanner MUST register one PTR record with the name: _uscan._tcp.local.

Scanners that support scans over a secure transport MUST advertise a SRV record in the form of <Scanner Name>._uscans._tcp.local. and SHOULD register a PTR record with the name: _uscans._tcp.local.

Clients adhering to eSCL 2.7 and above SHOULD NOT assume that _uscan service will always be supported by the scanner. This is to allow scanners that need to support only _uscans service for security reasons.

TXT Record Values

Scanners conforming to the eSCL Specification SHOULD provide the following keys in their TXT record for every eSCL scan service type supported - _uscan and _uscans. The key/value pairs MUST be populated unless explicitly mentioned as optional.

Key	Description
txtvers	Version of TXT record; SHOULD be 1. Example: txtvers=1 The txtvers SHOULD be the first key/value pair in the TXT record.
Vers	The eSCL interface version. If not specified by the scanner, the default version SHOULD be taken as 2.0. Example: vers=2.0

Key	Description
adminurl	configuration page URL for the Scanner. HTML human readable format. This key specifies the associated configuration URL for a given scanner. Example: adminurl=http://192.168.1.2/index.html The URL MAY have the scanner's hostname instead of the IP address. If the user changes the scanner's IP address or host name, the scanner SHOULD update the adminurl value~~.~~
representation	URL to a PNG or ICO file containing a graphical representation of the scanner. This key specifies a URL to a PNG or ICO file containing a graphical representation of the Scanner. If the URL points to a PNG file, the PNG file MUST be 128x128 in size and MUST contain an Alpha channel to mask the background surrounding the scanner. If the URL points to an ICO file (PREFERRED), the ICO file MUST at a minimum include 48x48, 128x128, and 512x512 representations of the scanner, and the various resolutions MUST be encoded using PNG, and MUST contain an Alpha channel to mask the background surrounding the scanner.
rs	resource portion of Scanner URI without leading slash. It communicates the 'root' URL for the eSCL interface. Example: rs=eSCL This field MUST reflect the {root} portion of all eSCL URLs. It does not have to be "eSCL", although "eSCL" is used in the rest of this document.
ty	human-readable make and model. The value of this key provides a user readable description of the make and model of the scanner which is suitable for display in a user interface when describing the scanner. Example: ty=HP Officejet Pro 8500

Key	Description
note	human-readable location. The value is a user readable location of the scanner which shows up in the Scan UI when browsing. Example: note=3rd Floor Copy Room The user SHOULD be able to specify the location of the scanner using the scanner configuration user interface. If no location information is available, the value of this key SHOULD be empty.
pdl	List of MIME media types supported by the scanner, separated by commas. Example: pdl=application/pdf,image/jpeg
uuid	A Universally Unique Identifier conforming to the RFC 4122. It identifies the device as a whole: if the scanner is a multifunction device, the same UUID must be used for the different functions: print, scan, fax, etc… For a given device, the UUID remains the same regardless of which I/O the eSCL transaction is coming from: Ethernet, Wifi, USB, etc… The UUID must have the same definition and be the same value as defined in the Bonjour specification. Example: uuid=96a4b400-2a9e-012f-6165-0025559efbc6f
cs	The ColorSpace defines the color capabilities of the scanner: “color” if the Scanner supports color scanning, “grayscale” if the scanner supports grayscale, “binary” if the scanner supports 1-bit monochrome scanning. The features are separated by commas.

Key	Description
	Example: cs=color,grayscale,binary
is	The InputSource defines the list of scan input options: "platen" for glass flat bed scanning, "adf" for Automatic Document Feeder, "camera" if the Scanner has a non-traditional scan bed (such as a stage); separated by commas. Example: is=platen,adf,camera Note that non-traditional scan beds are usually classified by the use of a camera (single shot capture) instead of by use of a fixed or movable scanning bar, hence the addition of the Camera keyword. Camera can be used both for situations where a camera (single) shot replaces the standard Scan mechanism (either from above or below the document) or for still or video cameras should they choose to implement this interface.
duplex	"T" if the Scanner has a duplex capable Automatic Document Feeder, otherwise "F" Example: duplex=T Note that Duplex is specifically an addition to the ADF flag, as it states that the ADF has a Duplexer. In other words it states that scanning two sides off of the ADF without user intervention is possible. A scanner can choose to prompt the user to scan the second side off of the Glass, a Simplex ADF, or with a Camera if it chooses regardless of the setting of the Duplex flag in the Bonjour TXT record.

4 eSCL Resources

The eSCL RESTful interface relies on the following resources:

- ScannerCapabilities – Scanner characteristics and Capabilities
- ScannerStatus – Scanner and job status (including asynchronous events)
- ScanBufferInfo – Estimate of scan page size
- ScanJobs – Factory to create new jobs
- ScanJob – Relates to a job instance management – limited to cancel in this version.
- ScanImageInfo – Actual size of the scan page
- ScanData – The scan data

ResourceType	Description	ResourceURI
ScannerCapabilities	Scanner capabilities and constraints	/{root}/ScannerCapabilities
ScannerStatus	Device and Job Status	/{root}/ScannerStatus
ScanBufferInfo	Scan settings validation and estimation	/{root}/ScanBufferInfo
ScanJobs	Factory element	/{root}/ScanJobs
ScanJob	Scan job instance	/{root}/ScanJobs/{job-id}
ScanImageInfo	Image info for a scan page	/{root}/ScanJobs/{job-id}/ScanImageInfo
ScanData	output image	URL is dynamic, specified at job creation

5 Intents

Often, mobile clients do not need extensive levels of control of the scanner engine, thus they don’t need to know about all the possible parameters a more complex scan application may require.

The objective of “intent” is:

the client application describes what it intends to do

the scanner handles the scan parameters internally

best-effort: the scanner assumptions behind the intent may not use the best possible rendering, but they are acceptable to the scan application.

The following intents are mandatory, all scanners MUST support them:

Mandatory Intents	Description
Document	Scanning optimized for text.
TextAndGraphic	A composite document with mixed text/graphic/photo content.
Photo	Scanning optimized for photo
Preview	Scanning optimized for performance (fast output)

Scanners MAY support additional intents (this list is not exhaustive):

Optional Intents	Description
Object	Scanning optimized for 3 dimensional objects - objects with depth
BusinessCard	Scanning optimized for a business card

All intents are advertised in ScannerCapabilities.

The scan application could provide additional (but optional) parameters alongside the intent:

- InputSource: platen, ADF, camera… If not specified, the scanner would use some heuristics to decide where to scan from - platen for photos - or use dedicated hardware - sheet sensor on ADF or platen.
- ScanRegion: if not specified, scanner uses edge detection if available; otherwise scan the largest scan-able area.
- ColorMode: for document, the client MAY specify using grayscale and binary monochrome modes.
- DocumentFormat: All scanners MUST support at least two formats (JPEG and PDF). Usually, by default, the intent drives which format is used: PDF for Text and TextAndGraphic, JPEG for Photo and Preview. However, if the scanner supports additional formats, the client can select a different format not conflicting with the intent.

Note: When an 'intent' is specified, the scanner is free to ignore extra parameters that are inconsistent with the intent, but SHOULD apply the settings that do not conflict with that intent's settings.

Based on the intent specified, the scanner MAY modify certain image processing related settings : DocumentFormatExt, ContentType, ColorSpace, Brightness, Contrast, Gamma, Highlight, NoiseRemoval, Shadow, Sharpen, Threshold etc., The other settings SHOULD be accepted by the scanner for the scan jobs.

For eg. if a scan client requests for a Document type intent with 600dpi resolution and the ContentType as Photo, the scanner SHOULD ignore the ContentType setting sent by the client and use Text instead. However the resolution of 600dpi SHOULD be accepted by the scanner even if 600dpi is not the default resolution.

6 Data Formats

All scanners and all eSCL clients MUST support at least two output data formats: JPEG and PDF. With PDF, the scanner can choose to embed JPEG and raster images within a PDF envelop. More high-end scanners could perform OCR before generating the PDF. Note that scanners can add 'Rotate' instructions (transforms) to PDF Page Objects if the page orientation is not correct.

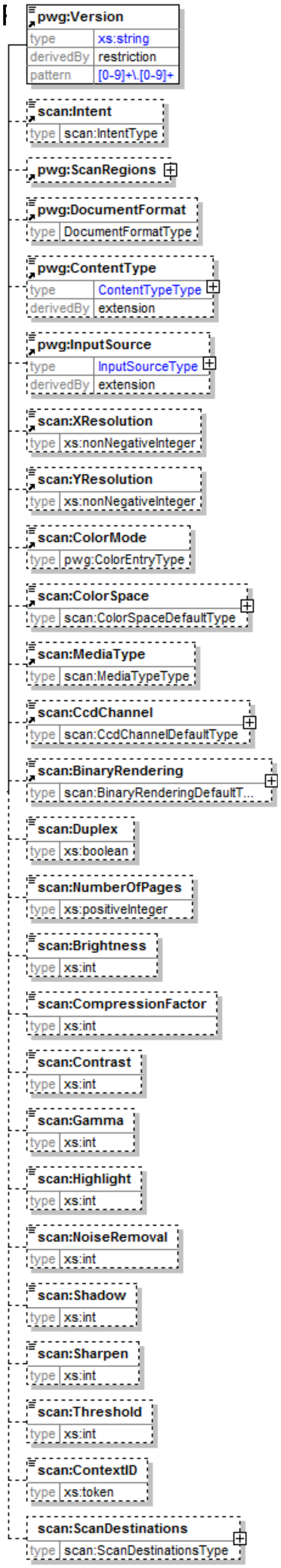
The scanner can support additional formats; they are advertised in ScannerCapabilities.

For all document formats that support multiple pages (PDF, TIFF etc.,), the scanner SHOULD upload a single file with all the scan pages merged together. For the document formats that do not support multiple pages (like JPEG), the clients SHOULD retrieve the individual JPEG scan images and can do any post-processing operation.

If clients need to save separate scan pages as PDFs, then they SHOULD request for JPEG scans and then convert them to PDF documents.

7 Scan Settings

All the scan settings are optional (except Version). When unspecified the scanner uses the default values advertised in the ScannerCapabilities resource. I



valid values for each element are also advertised in ScannerCapabilities. Some scanners MAY have sensors to detect the Input Source, Edges, Content Type, etc… If available, the scanner SHOULD use auto-detection or other available heuristics instead of default values.

- Version: only mandatory element. SHOULD be “2.0” or later versions.
- Intent: the type of scan the client application needs, or how the scan should be optimized for.
    - Document: scan optimized for text. It could be a multipage document.
    - TextAndGraphic: a composite document with mixed text/graphic/photo content
    - Photo: scan optimized for photo
    - Preview: low-quality high-speed mode
    - Object: object with 3-dimensional depth
- ScanRegions: list of scan regions. The number of supported concurrent regions is advertized in ScannerCapabilities. Scanners MUST support at least one region.
    - Each region defines:
        - Width and Height of the desired scan area. Expressed in the coordinate space.
        - XOffset and YOffset from the top-left origin point. Expressed in the coordinate space. XOffset+Width MUST be lesser or equal to MaxWidth (in ScannerCapabilities). YOffset + Height MUST be lesser or equal to MaxHeight (in ScannerCapabilities).
        - Units: MUST be “escl:ThreeHundredthsOfInches”
    - Attribute “MustHonor”: if set to ‘true’, the scanner SHOULD avoid edge detection. If ‘false’, or if attribute is not present the scanner can overwrite the requested scan region using embedded edge detection sensor or algorithm. In this case, the scan output MAY have the DNL marker to denote the actual height of the scanned image.
- DocumentFormat/ DocumentFormatExt: MIME type of the output image format. DocumentFormats is a collection of supported output formats the scanner can produce. eSCL 2.1 defines a new Document Format type (DocumentFormatExt) that allows a broader range of MIME type extension patterns which cannot be supported by the existing DocumentFormat element.
    - eSCL 2.1 providers MUST advertise BOTH element types in order to support eSCL 2.0 clients.
    - eSCL 2.1 consumers MUST use DocumentFormatExt values when submitting scan jobs to an eSCL 2.1 service/device because

DocumentFormat will be deprecated completely when support for eSCL 2.0 and prior eSCL versions ends.

- ContentType: Photo, Text, TextAndPhoto, LineArt, Magazine, Halftone or Auto (scanner choice: default setting or content type detection). Although similar to the 'Intent' element, this parameter is limited to image processing. Unlike the 'Intent' parameter, it does not imply default assumptions about other parameters (resolutions, data format, etc…). Especially, it SHOULD be used when the Intent isn't specified. It can be omitted when 'Intent' is present.
- InputSource:
  - Platen: a.k.a. glass flat bed
  - Feeder: a.k.a. ADF – Automatic Document Feeder
  - scan:Camera: object with depth scanning
- XResolution and YResolution: in DPI (dots per inch).
- Color Mode: combination of the mode (color, grayscale, binary black and white) with the bit depth.
  - BlackAndWhite1: binary monochrome scanning. Valid only for certain DocumentFormat/DocumentFormatExt values – like 'application/octet-stream', 'image/tiff' that can support single-bit scans. For document format not supporting BlackAndWhite1 color mode, scanner SHOULD report a 409 Conflict error.
  - Grayscale8: 8-bit grayscale
  - Grayscale16: 16-bit grayscale
  - RGB24: 8-bit per channel RGB
  - RGB48: 16-bit per channel RGB
- ColorSpace: sRGB is the likely option here. If not specified, the scanner uses the default value advertized in ScannerCapabilities.
- MediaType: if the scanner supports special media processing. Some media (paper or not) MAY require special handling.
- CcdChannel: indicates which CCD color channel to use for grayscale and monochrome scanning. It can also be a mix of all three channels. If not specified, the scanner uses the default value advertised in ScannerCapabilities.
  - Red: use the Red CCD
  - Green: use the Green CCD
  - Blue: use the Blue CCD
  - NTSC: weighted combination of the three color channels optimized for photos

  - GrayCcd: a dedicated Gray CCD array in the hardware (optimized for documents)
  - GrayCcdEmulated: an emulated Gray CCD mode where each CCD line are given even weight (1/3 R, 1/3 G, 1/3 B) (optimized for documents).

- BinaryRendering: Halftone or Threshold. If not specified, the scanner uses the default value advertized in ScannerCapabilities. This is used when the ColorMode is set to BlackAndWhite1.
- Duplex: Boolean – true means duplex is requested. It applies to hardware duplexer (most likely in ADF configuration). The scanner SHOULD ignore this parameter if it does not apply to the input source.
- NumberOfPages: if the ADF is able to pick detect and pick a predetermined number of pages (see capabilities attribute 'selectSinglePage'), the client can specify how many pages to scan.
- StoredJobRequest: Stored job requests are scan job requests sent to the scanner along with a job name to identify the request and an optional PIN to reserve the job. The job request is stored on the scanner without actually moving the scan bar. The actual scan job is started when user chooses the job name at the scanner control panel from the list of stored job requests and activates the stored job request by also keying in the PIN if sent by the client. With stored scan job requests, the documents are always with the user and hence the documents and the contents are safeguarded.

  When a user has chosen a particular stored job request for activation, the scanner should make sure to not allow any other scan jobs.

  The stored job requests can be tracked like any other scan job using the ScannerStatus. When the stored job request is not activated, the JobState should report as 'PendingHeld' and the JobStateReason as 'JobHeldByService'.

  The scanner MAY cancel the stored job request if user does not activate the job within a device-specific timeout if mentioned in the ScannerCapabilities xml. Alternatively, the client can cancel the job request by sending a DELETE request to the JobURI.

  - JobName: An identifier for the stored job request, so that this can be retrieved from a list of stored job requests at the control panel.This can be the host name of the client for easy identification by the user at the control panel. If more than one stored job request from a client has to be stored by the scanner at the same time, the client MUST send different job names to differentiate these job requests.
  - PIN: A secure PIN that needs to be keyed in by the user at the control panel to start the job. Admin can enforce that a stored job request with the PIN SHOULD be sent only over a secure connection. Admin can also enforce user credentials to be sent for this request. Refer to the Security section for more details.

- BlankPageDetection: Device can detect blank scan pages. If a blank page is detected, the page should still be returned to the client, but BlankPageDetected element of ScanImageInfo resource SHOULD be set appropriately.
- BlankPageDetectionAndRemoval: Device can detect blank scan pages and ignore them from sending them to the destination. When blank pages are detected by scanners that do not buffer the scan data, the scanner MAY wait for a longer time before sending the next page. If client requests for the NextDocument at this time, the scanner SHOULD reply with a 503 HTTP response along with the Retry-After header, to denote that the scanner is busy with processing and does not have a valid scan page ready to be sent. The client SHOULD resend the NextDocument request after the delay mentioned in the Retry-After header.
- OCR: For scanners that support optical character recognition, and can create a searchable PDF file. If set to true, the DocumentFormatExt setting SHOULD contain one of the supported PDF formats.
- OCRLanguage: The text language code as per ISO 639-1 that the OCR scanner should recognize within the scanned image.
- Brightness: this is used to adjust the brightness of the image. Min-Max ranges and default value are defined in Scan Capabilities. Lower values (than normal) give darker images; higher values give brighter images.
- CompressionFactor: integer. Lower numbers produce higher quality (at the expense of less compression). The absence of this field means that the scanner can decide the best compression/image-quality tradeoff, based on the intent (and/or other parameters).
- Gamma: the gamma value g used in the transform y=x^(1/g); Min-Max ranges and default value are defined in Scan Capabilities.
- Contrast: For adjusting the contrast. Min-Max ranges and default value are defined in Scan Capabilities. Lower values (than normal) generate less contrast; higher values generate more contrast.
- Highlight: The inflection point of image "highlights"; lower values lighten highlights.
- NoiseRemoval: defines the level of noise removal.
- Shadow: The inflection point of image "shadows"; lower values darken shadows.
- Sharpen: adjusts the sharpening level.
- Threshold: adjusts the threshold level for binary monochrome rendering.

For the above settings, Min-Max ranges and default value are defined in Scan Capabilities. Lower values attenuate the effect of the transform, whereas high values exacerbate them. If the field is not mentioned in the scan ticket, the scanner

SHOULD use the default value. If the value provided is outside the range, the scanner SHOULD use the closest valid value (most likely the Min or Max value).

- ContextID: opaque information relayed by the client.
- FeedDirection: Some scanners having a larger scannable area such as A3 MAY have trouble with the orientation of the smaller sized scans depending on how the media is positioned on the input source. For example, user can place a Letter-sized media on a A3 size scanbed in two different orientations - the long edge feed or the short edge feed. If the scanner is unable to edge detect, it is likely that the scanner produces a final image with the wrong orientation. Client can mention the feed direction of the media to avoid getting the final image in the wrong orientation.
    - LongEdgeFeed: Media is placed on the scanbed such that the longest edge is scanned first.
    - ShortEdgeFeed: Media is positioned such that the shortest edge is scanned first.

  Note that the client SHOULD take into consideration the feed direction while passing the Width and Height of the scan area. The Width in a LongEdgeFeed scan would be the Height in a ShortEdgeFeed scan and viceversa.

- EncryptedPdf: Scanner MAY support encryption of scanned PDF documents. User is allowed to configure the PDF version, encryption algorithm and the password for encryption.

  Any scan client request with the encryption password SHOULD be sent over a secure connection.

- ScanDestinations: Refer to Push Scan Jobs.

7.1 Coordinate Conventions

The document origin is coordinates 0-0 and is located at the arrow on the device marking the corner of the glass. For other input sources, the origin is located at the top-left equivalent of the scan area.

- For camera scanning, the X-axis is the first row of pixel.
- The X axis increases along the direction of the scanner sensor.
- The Y axis is in the direction of which the motor moves the scanner module. In ADF, it can be the direction of paper movement.
- Width is in the direction of the X axis. Height is in the direction of the Y axis.
- The coordinate system - used to describe the scan window – is defined in 1/300th of an inch points.

Note: *1/300th follows the TWAIN convention. We use 1/300th in the rest of this document.*

8 Capabilities

The capabilities resource provides the characteristics of the scanner. For each capability, the scanner provides which settings are available. The configuration resource is the union of the device capabilities. In one request, the client can get all relevant information about the scanner.

8.1 Data Formats

8.1.1 Resolution Capabilities

The scanner can define its resolution constraints by either defining a set of valid discrete resolutions, (exclusive) or by defining a range of valid values. Although it should not be a common case, the X and Y resolutions could be set differently.

- The scanner MAY define its resolution constraints for specific color modes, if the scan resolutions supported are different for every color mode.

Example of discrete resolutions:

```
<scan:SupportedResolutions>
  <scan:DiscreteResolutions>
    <scan:DiscreteResolution>
      <scan:XResolution>100</scan:XResolution>
      <scan:YResolution>100</scan:YResolution>
    </scan:DiscreteResolution>
    <scan:DiscreteResolution>
      <scan:XResolution>200</scan:XResolution>
      <scan:YResolution>200</scan:YResolution>
    </scan:DiscreteResolution>
    <scan:DiscreteResolution default=”true”>
      <scan:XResolution>300</scan:XResolution>
      <scan:YResolution>300</scan:YResolution>
    </scan:DiscreteResolution>
  </scan:DiscreteResolutions>
</scan:SupportedResolutions>
```

Example of a resolution range:

```
<scan:SupportedResolutions>
  <scan:ResolutionRange>
```

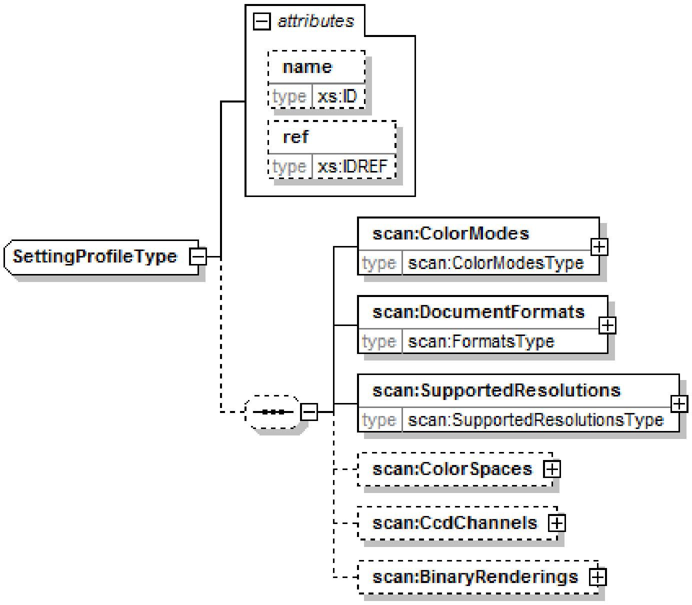
```
    <scan:XResolutionRange>
      <scan:Min>75</scan:Min>
      <scan:Max>1200</scan:Max>
      <scan:Normal>300</scan:Normal>
      <scan:Step>10</scan:Step>
    </scan:XResolutionRange>
    <scan:YResolutionRange>
      <scan:Min>75</scan:Min>
      <scan:Max>1200</scan:Max>
      <scan:Normal>300</scan:Normal>
      <scan:Step>10</scan:Step>
    </scan:YResolutionRange>
  </scan:ResolutionRange>
</scan:SupportedResolutions>
```

Example of discrete resolutions when dependent on the ColorMode:

```
<scan:SupportedResolutions>
      <scan:ColorMode>BlackAndWhite1</scan:ColorMode>
      <scan:DiscreteResolutions>
            <scan:DiscreteResolution>
                  <scan:XResolution>100</scan:XResolution>
                  <scan:YResolution>100</scan:YResolution>
            </scan:DiscreteResolution>
            <scan:DiscreteResolution>
                  <scan:XResolution>200</scan:XResolution>
                  <scan:YResolution>200</scan:YResolution>
            </scan:DiscreteResolution>
            <scan:DiscreteResolution>
                  <scan:XResolution>300</scan:XResolution>
                  <scan:YResolution>300</scan:YResolution>
            </scan:DiscreteResolution>
      </scan:DiscreteResolutions>
</scan:SupportedResolutions>
<scan:SupportedResolutions>
      <scan:ColorMode>RGB24</scan:ColorMode>
      <scan:DiscreteResolutions>
            <scan:DiscreteResolution>
                  <scan:XResolution>100</scan:XResolution>
                  <scan:YResolution>100</scan:YResolution>
            </scan:DiscreteResolution>
            <scan:DiscreteResolution>
                  <scan:XResolution>200</scan:XResolution>
                  <scan:YResolution>200</scan:YResolution>
            </scan:DiscreteResolution>
      </scan:DiscreteResolutions>
</scan:SupportedResolutions>
```

8.1.2 Setting Profile

A setting profile puts together the settings that are consistent with each other. It is a way to define constraints between scan attributes.



This construct can be used directly to define the set of settings or to reference an already defined profile.

- Color Mode: list of color modes (color, grayscale, binary monochrome) and bit depth.
- Document Formats: PDF, JPEG, etc…
    - DocumentFormats is a collection of supported output formats the scanner can produce. eSCL 2.1 defines a new Document Format type (DocumentFormatExt) that allows a broader range of MIME type extension patterns which cannot be supported by the existing DocumentFormat element.
    - Usage
        - eSCL 2.1 providers MUST advertise BOTH element types in order to support eSCL 2.0 clients.
        - eSCL 2.1 consumers MUST use DocumentFormatExt values when submitting scan jobs to an eSCL service/device because DocumentFormat will be deprecated completely when support for eSCL 2.0 ends.
- SupportedResolutions: options (range or discrete values)
- ColorSpaces: Available Color Spaces (defined in Scan Settings) and the default value.

- CcdChannels: Available CCD Channel options (defined in Scan Settings) for grayscale and monochrome scanning, and the default value.
- BinaryRenderings: Available Binary Rendering algorithms (defined in Scan Settings) and the default value.

The element can be defined and/or referenced from InputSourceCaps. For example, the following profile is reused by multiple xxxInputCaps, it can be defined as:

```
<scan:SettingProfile name="p1">
    <scan:ColorModes>
     <scan:ColorMode>BlackAndWhite1</scan:ColorMode>
    </scan:ColorModes>
    <scan:DocumentFormats>
     <!—For eSCL 2.0 and older clients→
     <pwg:DocumentFormat>application/pdf</pwg:DocumentFormat>
     <!—For eSCL 2.1 and newer clients →
     <scan:DocumentFormatExt>application/pdf</scan:DocumentFormatExt>
    </scan:DocumentFormats>
    <scan:SupportedResolutions>
     <scan:ResolutionRange>
       <scan:XResolutionRange>
        <scan:Min>75</scan:Min>
        <scan:Max>1200</scan:Max>
        <scan:Normal>300</scan:Normal>
       </scan:XResolutionRange>
       <scan:YResolutionRange>
        <scan:Min>75</scan:Min>
        <scan:Max>1200</scan:Max>
        <scan:Normal>300</scan:Normal>
       </scan:YResolutionRange>
     </scan:ResolutionRange>
    </scan:SupportedResolutions>
    <scan:CcdChannels>
     <scan:CcdChannel 
scan:default="true">GrayCcdEmulation</scan:CcdChannel>
    </scan:CcdChannels>
    <scan:BinaryRenderings>
     <scan:BinaryRendering>Threshold</scan:BinaryRendering>
     <scan:BinaryRendering 
scan:default="true">Halftone</scan:BinaryRendering>
    </scan:BinaryRenderings>
   </scan:SettingProfile>
```

Then, it can be referenced from any xxxInputCaps:

```
<scan:AdfSimplexInputCaps>
 ...
 <scan:SettingProfiles>
  <scan:SettingProfile ref="p1"/>
 </scan:SettingProfiles>
 ...
</scan:AdfSimplexInputCaps>
```

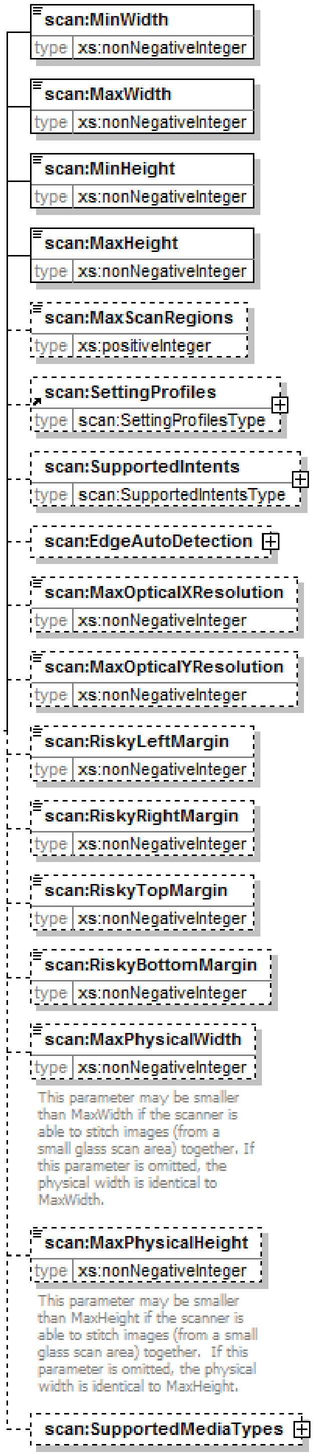
If a SettingProfile is used only once, it can be directly embedded in a xxxInputCaps

```
   <scan:PlatenInputCaps>
    ...
    <scan:SettingProfiles>
     <scan:SettingProfile>
      <scan:ColorModes>
       <scan:ColorMode>RGB24</scan:ColorMode>
       <scan:ColorMode>Grayscale8</scan:ColorMode>
       <scan:ColorMode>BlackAndWhite1</scan:ColorMode>
      </scan:ColorModes>
      <scan:DocumentFormats>
      <!—For eSCL 2.0 and older clients→
       <pwg:DocumentFormat>image/jpeg</pwg:DocumentFormat>
       <pwg:DocumentFormat>application/pdf</pwg:DocumentFormat>
      <!—For eSCL 2.1 and newer clients →
       <scan:DocumentFormatExt>image/jpeg</scan:DocumentFormatExt>

<scan:DocumentFormatExt>application/pdf</scan:DocumentFormatExt>
      </scan:DocumentFormats>
    <scan:SupportedResolutions>
     <scan:ResolutionRange>
      <scan:XResolutionRange>
       <scan:Min>75</scan:Min>
       <scan:Max>1200</scan:Max>
       <scan:Normal>300</scan:Normal>
      </scan:XResolutionRange>
      <scan:YResolutionRange>
       <scan:Min>75</scan:Min>
       <scan:Max>1200</scan:Max>
       <scan:Normal>300</scan:Normal>
      </scan:YResolutionRange>
     </scan:ResolutionRange>
    </scan:SupportedResolutions>
      <scan:ColorSpaces>
```

```
        <scan:ColorSpace scan:default="true">sRGB</scan:ColorSpace>
       </scan:ColorSpaces>
       <scan:CcdChannels>
        <scan:CcdChannel>Red</scan:CcdChannel>
        <scan:CcdChannel>Blue</scan:CcdChannel>
        <scan:CcdChannel>Green</scan:CcdChannel>
        <scan:CcdChannel scan:default="true">NTSC</scan:CcdChannel>
       </scan:CcdChannels>
       <scan:BinaryRenderings>
        <scan:BinaryRendering
scan:default="true">Halftone</scan:BinaryRendering>
        <scan:BinaryRendering>Threshold</scan:BinaryRendering>
       </scan:BinaryRenderings>
     </scan:SettingProfile>
    </scan:SettingProfiles>
</scan:PlatenInputCaps>
```

8.1.3 InputSourceCaps

InputSourceCaps are specified for each input source (Platen, ADF and ADF Duplex).


scan:MinWidth
type xs:nonNegativeInteger
scan:MaxWidth
type xs:nonNegativeInteger
scan:MinHeight
type xs:nonNegativeInteger
scan:MaxHeight
type xs:nonNegativeInteger
scan:MaxScanRegions
type xs:positiveInteger
scan:SettingProfiles
type scan:SettingProfilesType
scan:SupportedIntents
type scan:SupportedIntentsType
scan:EdgeAutoDetection
scan:MaxOpticalXResolution
type xs:nonNegativeInteger
scan:MaxOpticalYResolution
type xs:nonNegativeInteger
scan:RiskyLeftMargin
type xs:nonNegativeInteger
scan:RiskyRightMargin
type xs:nonNegativeInteger
scan:RiskyTopMargin
type xs:nonNegativeInteger
scan:RiskyBottomMargin
type xs:nonNegativeInteger
scan:MaxPhysicalWidth
type xs:nonNegativeInteger
This parameter may be smaller than MaxWidth if the scanner is able to stitch images (from a small glass scan area) together. If this parameter is omitted, the physical width is identical to MaxWidth.
scan:MaxPhysicalHeight
type xs:nonNegativeInteger
This parameter may be smaller than MaxHeight if the scanner is able to stitch images (from a small glass scan area) together. If this parameter is omitted, the physical width is identical to MaxHeight.
scan:SupportedMediaTypes

- MaxWidth: Maximum width that can be scanned in coordinates specified by the coordinate space.
- MinWidth: Minimum width that can be scanned in coordinates specified by the coordinate space.
- MaxHeight: Maximum height that can be scanned in coordinates specified by the coordinate space
- MinHeight: Minimum height that can be scanned in coordinates specified by the coordinate space
- MaxXOffset, MaxYOffset: Maximum offsets supported from the top-left origin point. All scanners supporting eSCL ver 2.9 or later MUST report these elements if supporting offsets. If the scanner does not support an offset, the scanner MAY indicate a value of 0 for that offset.

  Note: The scanner MAY support any one of these offsets independent of the other. The unsupported offsets are either omitted or SHOULD have the value of 0.
- MaxScanRegions: How many concurrent scan regions supported. Default is 1 when the field is not present.
- SettingProfiles: list of Setting Profiles. They are either defined directly here, or they reference Setting Profiles defined in the ScannerCapabilities. Device MUST advertise only one SettingProfile for a particular input source.
- SupportedIntents: list of intents supported by this input source.
- EdgeAutoDetection: list of which edge(s) the scanner can automatically detect:
  - TopEdge
  - LeftEdge
  - BottomEdge
  - RightEdge
- MaxOpticalXResolution: Maximum Optical Resolution device can support in X axis. The units are PPI (pixels per inch).

- MaxOpticalYResolution: Maximum Optical Resolution device can support in Y axis. The units are PPI (pixels per inch).
- Risky…Margins: if the scan output quality diminishes near the edges of the scan area, the risky margins identify the area at risk: Left, Right, Top, Bottom margins from the edge.
- MaxPhysicalWidth and MaxPhysicalHeight: in case the physical dimensions of the scan area are different from MaxWidth and MaxHeight. Possible if scanner is able to stitch multiple images into a bigger image.
- SupportedMediaTypes: Only applies to scanners that can perform special processing based on the type of object or media put on the input source.
  - Object-3D: applies to camera based scanner
  - Blueprint, OldRecycled, Translucent,…: paper types that MAY require special scanning options.
- FeedDirections: Applies to scanners that can perform scanning with any feed direction of the input media.
  - LongEdgeFeed/ShortEdgeFeed: Media is placed such that the longest/shortest edge of the media is scanned first.

8.1.3.1 Edge Detection

Note that for ADF scanning with JPEG output, the scanner MAY only be able to detect the page height (aka length) at the end of page scan. Because the width and height information is at the beginning of the JPEG file format, the scanner SHOULD be able to scan and store the whole page before sending the JPEG payload.

If the scanner cannot detect the page height before starting the scan and if it cannot store the whole page, the scanner SHOULD ignore the end of page sensor information and instead honor the scan height parameter in the scan job settings. If the requested height is longer than the real document's length, the scanner MUST handle the condition gracefully: for example, pad the image to the requested length.

If the scanner can edge detect, then the scanner SHOULD provide the actual width and height to the clients through the ScanImageInfo – explained later in the section Scan Job. (The ScanImageInfo support is available from eSCL2.3 onwards).

8.1.3.2 Output Orientation

Some duplex ADF scanners MAY scan the reverse (even) page upside down, it is up to the scanner to correct that behavior by rotating the page. As a general rule, whatever output format the user selects (JPEG, PDF,…), the scanner SHOULD generate output file format that can be directly consumed by the end user, without extra transformations: e.g. no JPEG DNL trailer for height, no need to flip or rotate images or pages.

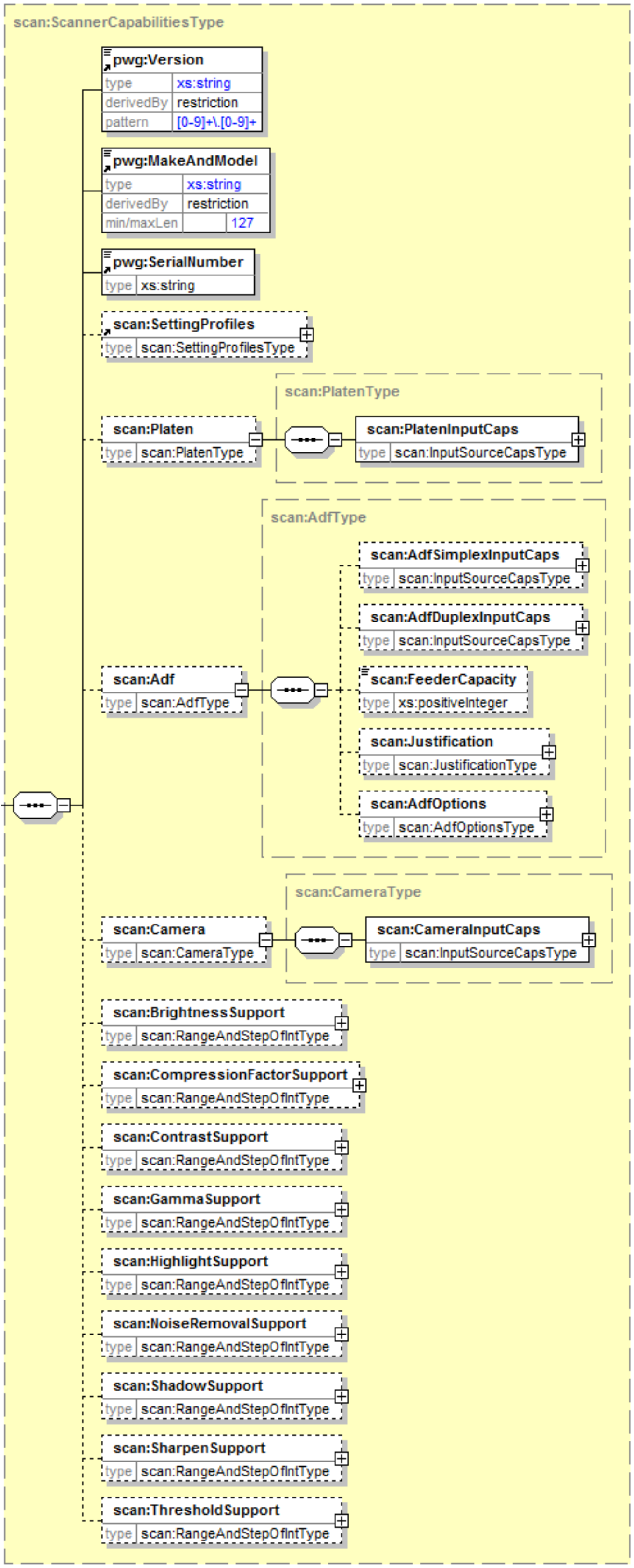
PDF allows clients to perform ‘Rotate’ transforms on PDF Page Objects to compensate scanners limitations. Note that JPEG is less forgiving…

8.1.4 Capabilities

- Version: each eSCL resource MUST specify the interface version number
- MakeAndModel: the scanner maker and model name
- SerialNumber: serial number uniquely identifying the device

— Manufacturer: the IEEE-1284 Device ID MANUFACTURER/MFG string. This element SHALL be the same as that provided in the Bonjour TXT record discovery key “usb_MFG”.

- Certifications: the list of certifications and their versions as defined by the respective certification specifications.
- UUID: A Universally Unique Identifier conforming to RFC 4122. This MUST be the same value reported in the uuid Bonjour TXT record.
- AdminURI: Configuration page URL for the Scanner. This MUST be the same value as the adminurl Bonjour TXT record.
- IconURI: URL to an image file containing graphical representation of the scanner. This MUST be the same value as the representation Bonjour TXT record.



  - SettingProfiles: list of Setting Profiles that are referenced by PlatenInputCaps, AdfSimplexInputCaps, AdfDuplexInputCaps or CameraInputCaps. Because SettingProfiles can also be defined directly in the different xxxInputCaps, this list is used for SettingProfiles that are common to (and referenced from) multiple xxxInputCaps.
  - Platen
    - InputSourceCaps for a scan job from glass
  - ADF
    - AdfSimplexInputCaps: the InputSourceCaps for the ADF in simplex mode
    - AdfDuplexInputCaps: the InputSourceCaps for the ADF in duplex mode. This element is optional: omitted if ADF does not support duplex.
    - FeederCapacity: maximum number of pages the ADF can contain.
    - Justification: specify how the ADF justify the document.
      - XImagePosition: Left/Center/Right for the width
      - YImagePosition: Top/Center/Bottom for the height
  - AdfOptions:
    - DetectPaperLoaded: device can detect if paper is loaded in the tray
    - SelectSinglePage: ADF can scan one page if requested, even if additional pages are in the ADF. Most of our ADFs do not have this capability and scan until the ADF is empty. If the ADF does not support SelectSinglePage and the app asks for fewer pages than in the ADF; we will need some way of cleaning up the job…
    - Duplex support
  - Camera
    - InputSourceCaps associated with the camera scanner
- StoredJobRequestSupport: Reports if device can support stored scan job requests.
  - MaxStoredJobRequests: Gives the maximum number of stored job requests that the scanner can handle.
  - TimeoutInSeconds: The timeout for the stored job requests. If the job is not activated within this time period, then the job request MUST be deleted by the scanner.
  - PINLength: The length of the PIN that can be entered on the device front panel. This should be the actual length of the PIN passed by the client for

a stored job request. Scanner SHOULD support a minimum of 4 characters for the PIN.

  - MaxJobNameLength: Maximum length of the stored job request name that the device can handle.

- BlankPageDetection: Device can detect blank scan pages.

- BlankPageDetectionAndRemoval: Device can detect blank scan pages and ignore them from sending them to the destination

- OCRSupport: Device can create a searchable PDF file as the scan output

- OCRLanguageSupport: ISO 639-1 language codes that device can recognize within the scanned image.

- Image transform range support: BrightnessSupport, CompressionFactorSupport, etc…

- These elements define the minimum, maximum, average/normal (and optionally the step) values for available transforms. Omission of the element indicates that the corresponding transform is not supported.

- EncryptedPdfSupport: Scanner MAY support scanning of encrypted PDF documents.

8.2 Interface

Scan Capabilities			
/{root}/ScannerCapabilities			
GET	Description	List the scanner capabilities: color entries, scan modes, ADF,…	
	Payload	OUT	XML: /scan:ScannerCapabilities
	Status Code	200 OK – Success 500 Internal Server Error - Unknown internal error. 503 Service Unavailable – Server busy. Retry later	

All examples in this document assume that the scanner is network scanner with the IP address 192.168.0.100.

The examples have been kept to a strict minimum, avoiding the complex settings.

8.3 Usage Flow

➔

```
GET /eSCL/ScannerCapabilities HTTP/1.1
Host: 192.168.0.100
```

🡐

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 4606

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScannerCapabilities
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03"
xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <pwg:MakeAndModel>Hewlett-Packard Photosmart C4760</pwg:MakeAndModel>
  <pwg:SerialNumber>CN017971874378PJ</pwg:SerialNumber>
  <scan:UUID>96a4b400-2a9e-012f-6165-0025559efbc6f</scan:UUID>
  <scan:AdminURI>http://192.168.1.2/index.html</scan:AdminURI>
  <scan:IconURI>http://192.168.1.2/scanner.png</scan:IconURI>
  <scan:SettingProfiles>
    <scan:SettingProfile name="p1">
     <scan:ColorModes>
       <scan:ColorMode>BlackAndWhite1</scan:ColorMode>
       <scan:ColorMode>Grayscale8</scan:ColorMode>
     </scan:ColorModes>
     <scan:DocumentFormats>
       <pwg:DocumentFormat>application/pdf</pwg:DocumentFormat>
       <pwg:DocumentFormat>image/jpeg</pwg:DocumentFormat>
      <scan:DocumentFormatExt>application/pdf</scan:DocumentFormatExt>
      <scan:DocumentFormatExt>image/jpeg</scan:DocumentFormatExt>
     </scan:DocumentFormats>
     <scan:SupportedResolutions>
       <scan:DiscreteResolutions>
        <scan:DiscreteResolution>
        <scan:XResolution>100</scan:XResolution>
        <scan:YResolution>100</scan:YResolution>
        </scan:DiscreteResolution>
        <scan:DiscreteResolution>
        <scan:XResolution>200</scan:XResolution>
        <scan:YResolution>200</scan:YResolution>
        </scan:DiscreteResolution>
        <scan:DiscreteResolution>
        <scan:XResolution>300</scan:XResolution>
        <scan:YResolution>300</scan:YResolution>
        </scan:DiscreteResolution>
       </scan:DiscreteResolutions>
     </scan:SupportedResolutions>
     <scan:CcdChannels>
       <scan:CcdChannel>Red</scan:CcdChannel>
       <scan:CcdChannel scan:default="true">Blue</scan:CcdChannel>
     </scan:CcdChannels>
     <scan:BinaryRenderings>
       <scan:BinaryRendering>Threshold</scan:BinaryRendering>
       <scan:BinaryRendering
```

```
scan:default="true">Halftone</scan:BinaryRendering>
     </scan:BinaryRenderings>
    </scan:SettingProfile>
  </scan:SettingProfiles>
  <scan:Platen>
    <scan:PlatenInputCaps>
     <scan:MinWidth>1</scan:MinWidth>
     <scan:MaxWidth>3000</scan:MaxWidth>
     <scan:MinHeight>1</scan:MinHeight>
     <scan:MaxHeight>3600</scan:MaxHeight>
     <scan:MaxScanRegions>2</scan:MaxScanRegions>
     <scan:SettingProfiles>
       <scan:SettingProfile>
        <scan:ColorModes>
        <scan:ColorMode>RGB24</scan:ColorMode>
        <scan:ColorMode>Grayscale8</scan:ColorMode>
        <scan:ColorMode>BlackAndWhite1</scan:ColorMode>
        </scan:ColorModes>
        <scan:DocumentFormats>
        <pwg:DocumentFormat>image/jpeg</pwg:DocumentFormat>
        <pwg:DocumentFormat>application/pdf</pwg:DocumentFormat>
        <scan:DocumentFormatExt>image/jpeg</scan:DocumentFormatExt>
        <scan:DocumentFormatExt>application/pdf</scan:DocumentFormatExt>
        </scan:DocumentFormats>
        </scan:DocumentFormats>
        <scan:SupportedResolutions>
        <scan:ResolutionRange>
          <scan:XResolutionRange>
           <scan:Min>75</scan:Min>
           <scan:Max>1200</scan:Max>
           <scan:Normal>300</scan:Normal>
           <scan:Step>10</scan:Step>
          </scan:XResolutionRange>
          <scan:YResolutionRange>
           <scan:Min>75</scan:Min>
           <scan:Max>1200</scan:Max>
           <scan:Normal>300</scan:Normal>
           <scan:Step>10</scan:Step>
          </scan:YResolutionRange>
        </scan:ResolutionRange>
        </scan:SupportedResolutions>
        <scan:ColorSpaces>
        <scan:ColorSpace scan:default="true">sRGB</scan:ColorSpace>
        </scan:ColorSpaces>
        <scan:CcdChannels>
        <scan:CcdChannel>Red</scan:CcdChannel>
        <scan:CcdChannel>Blue</scan:CcdChannel>
        <scan:CcdChannel>Green</scan:CcdChannel>
        <scan:CcdChannel scan:default="true">NTSC</scan:CcdChannel>
        </scan:CcdChannels>
        <scan:BinaryRenderings>
        <scan:BinaryRendering
scan:default="true">Halftone</scan:BinaryRendering>
        <scan:BinaryRendering>Threshold</scan:BinaryRendering>
        </scan:BinaryRenderings>
       </scan:SettingProfile>
```

```
        <scan:SettingProfile ref="p1"/>
      </scan:SettingProfiles>
      <scan:MaxOpticalXResolution>1200</scan:MaxOpticalXResolution>
      <scan:MaxOpticalYResolution>1200</scan:MaxOpticalYResolution>
      <scan:RiskyLeftMargin>28</scan:RiskyLeftMargin>
      <scan:RiskyRightMargin>30</scan:RiskyRightMargin>
      <scan:RiskyTopMargin>32</scan:RiskyTopMargin>
      <scan:RiskyBottomMargin>44</scan:RiskyBottomMargin>
  </scan:PlatenInputCaps>
  </scan:Platen>
  <scan:Adf>
    <scan:AdfSimplexInputCaps>
      <scan:MinWidth>1</scan:MinWidth>
      <scan:MaxWidth>2600</scan:MaxWidth>
      <scan:MinHeight>1</scan:MinHeight>
      <scan:MaxHeight>3400</scan:MaxHeight>
      <scan:SettingProfiles>
        <scan:SettingProfile ref="p1"/>
      </scan:SettingProfiles>
      <scan:EdgeAutoDetection>
        <scan:SupportedEdge>BottomEdge</scan:SupportedEdge>
      </scan:EdgeAutoDetection>
      <scan:MaxOpticalXResolution>300</scan:MaxOpticalXResolution>
      <scan:MaxOpticalYResolution>300</scan:MaxOpticalYResolution>
      <scan:RiskyLeftMargin>28</scan:RiskyLeftMargin>
      <scan:RiskyRightMargin>30</scan:RiskyRightMargin>
      <scan:RiskyTopMargin>32</scan:RiskyTopMargin>
      <scan:RiskyBottomMargin>44</scan:RiskyBottomMargin>
    </scan:AdfSimplexInputCaps>
    <scan:FeederCapacity>100</scan:FeederCapacity>
    <scan:AdfOptions>
      <scan:AdfOption>DetectPaperLoaded</scan:AdfOption>
      <scan:AdfOption>SelectSinglePage</scan:AdfOption>
    </scan:AdfOptions>
  </scan:Adf>
  <scan:StoredJobRequestSupport>
    <scan:MaxStoredJobRequests>10</scan:MaxStoredJobRequests>
    <scan:TimeoutInSeconds>120</scan:TimeoutInSeconds>
  </scan:StoredJobRequestSupport>
  <scan:BlankPageDetection>true</scan:BlankPageDetection>
  <scan:BlankPageDetectionAndRemoval>true</scan:BlankPageDetectionAndRem
oval>
  <scan:OCRSupport>true</scan:OCRSupport>
  <scan:OCRLanguageSupport>
    <pwg:NaturalLanguageSupported>ja</pwg:NaturalLanguageSupported>
    <pwg:NaturalLanguageSupported>fr</pwg:NaturalLanguageSupported>
  </scan:OCRLanguageSupport>
</scan:ScannerCapabilities>
```

9 Status

The scanner status SHOULD be used to:

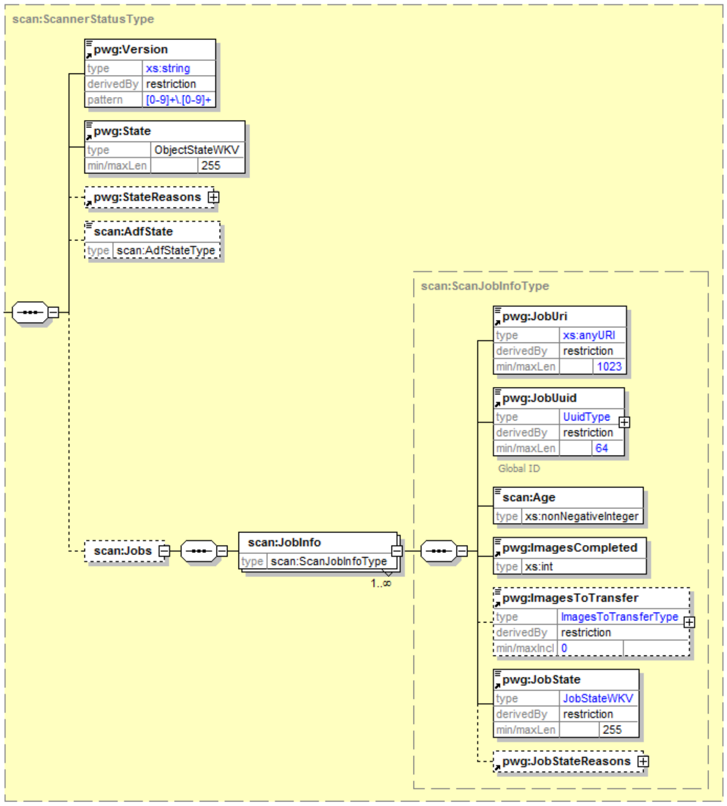
- retrieve the scanner status before requesting a scan job
- check if the scanner is available
- check if media is loaded in ADF.
- check if a scan job is active or canceled

Scan Jobs can be sorted chronologically using the Age element. Note that scanners are NOT required to provide the jobs elements in chronological order in the XML structure.

The most recent job element allows the client app to detect if the scan upload was successful.

For example, when getting a single page scan, the scanner MAY return some data, but the data may be incomplete or truncated (e.g. the connection is dropped).

9.1 Data Format



JobInfo reports the state of a particular scan job:

- JobUuid: Job unique identifier. This element MUST remain unique over power-cycles.
- JobUri: each job is given a URL that identifies the job. Note that the URL uniqueness is not guaranteed over power-cycles. The JobUri is returned by the scanner when the job is created. For scanners supporting eSCL 2.6 and above, the JobUri SHALL contain a Universally Unique Identifier that conforms to the RFC 4122.
- Age: time in seconds since the job info has been updated. The duration is the difference between the time of the latest update to job info element relative to the time of the status request.
- ImagesCompleted: the number of images or pages scanned up to the time of the status request. For the active job, this field can change (increment) in further requests.

- ImagesToTransfer: The scanner MAY have stored a few pages that have not been transferred yet. Not all scanners are able to support this feature. This element SHOULD NOT be used for push scans.
- JobState:
  - Canceled: end state – indicates that the job was canceled either by the remote client application (thru the eSCL interface) or by the user interacting with the scanner directly. Check JobStateReasons for more details.
  - Aborted: end state – either an internal device error, or a communication error or a security error
  - Completed: job is finished successfully
  - Pending: the job was initiated, and the scanner is preparing the scan engine
  - Processing: the scanner is processing the job and is transmitting the scan data

ScannerState reports the overall state of the scanner (as opposed to the scan job state):

- Idle
- Processing: busy with some job or activity
- Testing: calibrating, preparing the unit
- Stopped: error condition occurred
- Down: unit is unavailable

Check StateReasons and AdfState for more information

AdfState reports ADF specific states:

- ScannerAdfProcessing – the OK state, other states are errors or require ‘user attention’.
- ScannerAdfEmpty – scanner ADF is not loaded with sheets
- ScannerAdfJam – the scan sheet is jammed in the ADF
- ScannerAdfLoaded – ADF is loaded with sheets
- ScannerAdfMispick – the scanner not able to pick a sheet
- ScannerAdfHatchOpen – the ADF hatch is open
- ScannerAdfDuplexPageTooShort – the sheet placed in ADF is shorter than the minimum size for a duplex scan
- ScannerAdfDuplexPageTooLong – the sheet placed in ADF is longer than the max allowed size for a duplex scan
- ScannerAdfMultipickDetected – scanner has picked multiple sheets instead of one sheet
- ScannerAdfInputTrayFailed – ADF tray failure
- ScannerAdfInputTrayOverloaded – ADF tray is stacked with more sheets than its capacity

The list of JobInfo shows the current active or completed scan job, and at least one previous job. All scanners MUST support at least 2 entries. This element isn't persistent over power cycles.

9.2 Interface

Scanner Status			
/{root}/ScannerStatus			
GET	Description	Returns the current state of the scanner, including active and last job states.	
	Payload	OUT	XML: /scan:ScannerStatus
	Status Code	200 OK – Success 301 Moved - The client is redirected to a secure connection. 401 Unauthorized - The client is challenged for access credentials. 500 Internal Server Error - Unknown internal error. 503 Service Unavailable – Server busy. Retry later	

9.3 Usage Flow

Note that the scanner is not required to use the namespace prefix 'scan'. The client application MUST honor any prefix.

➔

```
GET /eSCL/ScannerStatus HTTP/1.1
Host: 192.168.0.100
```

🡐

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 1158


<?xml version="1.0" encoding="UTF-8"?>
<scan:ScannerStatus
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03"
xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <pwg:State>Processing</pwg:State>
  <scan:Jobs>
```

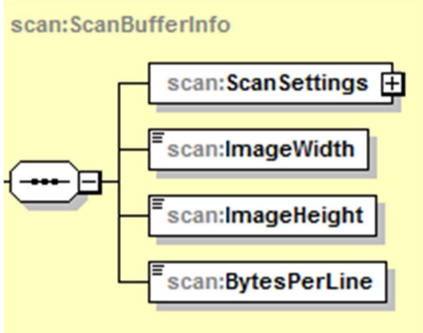
```
   <scan:JobInfo>
    <pwg:JobUri>/ScanJobs/893e6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUri>
    <pwg:JobUuid>893e6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUuid>
    <scan:Age>10</scan:Age>
    <pwg:ImagesCompleted>1</pwg:ImagesCompleted>
    <pwg:ImagesToTransfer>1</pwg:ImagesToTransfer>
    <pwg:JobState>Processing</pwg:JobState>
    <pwg:JobStateReasons>
     <pwg:JobStateReason>JobScanning</pwg:JobStateReason>
    </pwg:JobStateReasons>
   </scan:JobInfo>
   <scan:JobInfo>
    <pwg:JobUri>/ScanJobs/898d6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUri>
    <pwg:JobUuid>898d6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUuid>
    <scan:Age>220</scan:Age>
    <pwg:ImagesCompleted>5</pwg:ImagesCompleted>
    <pwg:ImagesToTransfer>0</pwg:ImagesToTransfer>
    <pwg:JobState>Completed</pwg:JobState>
    <pwg:JobStateReasons>

  <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobState
Reason>
    </pwg:JobStateReasons>
   </scan:JobInfo>
  </scan:Jobs>
</scan:ScannerStatus>
```

10 Scan Buffer Info

This resource is used to provide the clients with an estimation of scan size for a given scan settings. This is used by clients to calculate the needed buffer sizes to hold the subsequent scan. The estimation values MAY be different from the actual scan size, when device can do edge detection. This resource can also be used to validate the scan settings.

10.1 Data Format

- ImageWidth, ImageHeight: Device estimated width and height in pixels for a scan page. This is calculated before the start of the scan job for the given input settings, so actual values can be different.
- BytesPerLine: Number of bytes required for one row or line of raw uncompressed raster image.
- ScanSettings: Device modified settings during validation. Device can return an error or can change settings so that a scan can be performed with the modified settings..



10.2 Interface

Scan Buffer Info			
/{root}/ScanBufferInfo			
PUT	Description	Submit a set of scan settings, and expect the server to validate the settings and return an estimation of the future scan image size. Note that this resource is available outside a job context. This query should work even if the scanner is processing a job from a different application or client computer	
	Payload	IN	XML: /scan:ScanSettings. The client only sends the proposed scan settings
		OUT	XML: /scan:ScanBufferInfo. The server returned negotiated scan settings and the scan image size estimation
	Status Code	200 OK – Success	
		400 Bad Request – Request not understood due to wrong syntax	
		409 Conflict – Invalid input	

		500 Internal Server Error - Unknown internal error.
		503 Service Unavailable – Server busy. Retry later

10.3 Usage Flow

➔

```
PUT /eSCL/ScanBufferInfo HTTP/1.1
Host: 192.168.0.100
Content-Type: text/xml
Content-Length: 637

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScanSettings
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <scan:Intent>Photo</scan:Intent>
  <pwg:ScanRegions>
   <pwg:ScanRegion>
     <pwg:Height>1200</pwg:Height>

  <pwg:ContentRegionUnits>escl:ThreeHundredthsOfInches</pwg:ContentRegionUnits>
     <pwg:Width>1800</pwg:Width>
     <pwg:XOffset>0</pwg:XOffset>
     <pwg:YOffset>0</pwg:YOffset>
   </pwg:ScanRegion>
  </pwg:ScanRegions>
  <pwg:InputSource>Platen</pwg:InputSource>
  <scan:ColorMode>Grayscale8</scan:ColorMode>
</scan:ScanSettings>
```

🡐

```
HTTP/1.1 200 OK
Content-Type: text/xml
```

```
Content-Length: 1158


<?xml version="1.0" encoding="UTF-8"?>
<scan:ScanBufferInfo
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03
"
xmlns:httpdest="http://schemas.hp.com/imaging/httpdestina
tion/2011/10/13"
xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <scan:ScanSettings>
   <pwg:Version>2.6</pwg:Version>
   <scan:Intent>Photo</scan:Intent>
   <pwg:ScanRegions>
    <pwg:ScanRegion>
     <pwg:Height>1200</pwg:Height>

  <pwg:ContentRegionUnits>escl:ThreeHundredthsOfInches</p
wg:ContentRegionUnits>
     <pwg:Width>1800</pwg:Width>
     <pwg:XOffset>0</pwg:XOffset>
     <pwg:YOffset>0</pwg:YOffset>
    </pwg:ScanRegion>         </pwg:ScanRegions>

  <scan:DocumentFormatExt>image/jpeg</scan:DocumentFormat
Ext>
   <pwg:ContentType>Photo</pwg:ContentType>
   <pwg:InputSource>Platen</pwg:InputSource>
   <scan:XResolution>300</scan:XResolution>
   <scan:YResolution>300</scan:YResolution>
   <scan:ColorMode>Grayscale8</scan:ColorMode>
   <scan:ColorSpace>YCC</scan:ColorSpace>
   <scan:CcdChannel>GrayCcdEmulated</scan:CcdChannel>
   <scan:BinaryRendering>Threshold</scan:BinaryRendering>
  </scan:ScanSettings>
  <scan:ImageWidth>1200</scan:ImageWidth>
  <scan:ImageHeight>1800</scan:ImageHeight>
  <scan:BytesPerLine>1500</scan:BytesPerLine>
</scan:ScanBufferInfo>
```

11 Scan Job

11.1 Description

After initiating the scan job, the client is responsible to pace the retrieval of the scan pages. In this document, this transactional model is labeled: Pull Scan.

Once the job is created, the scanner SHOULD be in Pending or Processing state. The client keeps checking the scanner status and the client uses the job URL to upload the next scan page after the JobState moves to Pending or Processing. The client keeps invoking this URL until the NextDocument request returns 404 Not Found. When all pages are scanned, the JobState MUST indicate ‘Completed’ denoting there are no more pages to be uploaded. The JobState SHOULD indicate ‘Canceled’ or ‘Aborted’ if the job was either canceled by the user or due to internal abort

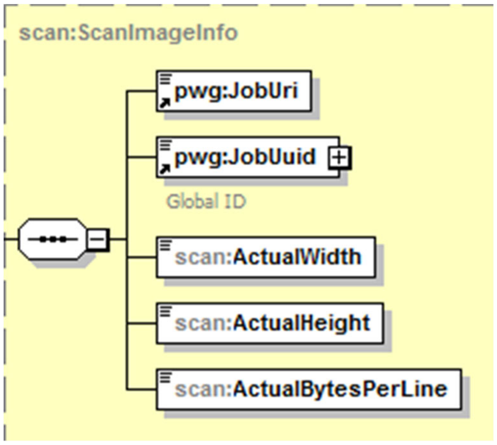
If the NextDocument request times out, the client SHOULD check the JobState and retry pulling the scan pages if the JobState is either Pending or Processing.

After the job is created, if the client fails to retrieve the scan pages for some reason – could be even a network failure, the scanner MAY abort the scan job after a certain time period determined by the scanner. The scanner SHOULD NOT cancel the job if it can retain the scan pages for a longer period without resource constraints.

11.2 Scan Image Info

For some scan jobs, especially a raw scan job, the scan clients MAY want to know the actual scan width and height if the scanner has edge detection. In this case, the scanner SHOULD provide the actual width and height values to the clients through the ScanImageInfo (from eSCL2.3 onwards).

A client MAY retrieve the ScanImageInfo for every scan page, immediately after successfully retrieving the scan page. Some clients MAY need to render the scan data after receiving every chunk of scan data. The device SHOULD support these clients by sending back the ScanImageInfo, whether the client requests for this info after sending the first chunk or after sending the entire page.



- ActualWidth, ActualHeight: Actual width and height of the scanned image detected by sensors, if present. These size parameters MAY be different from the estimated ScanBufferInfo values, especially for an ADF scan job. Devices that have only the height sensors SHOULD report the ActualWidth to be the same as the estimated ImageWidth in ScanBufferInfo.

- ActualBytesPerLine: Number of bytes required for one row or line of raw uncompressed raster image.
- BlankPageDetected: A Boolean value. Tells if a blank page was detected. Valid if BlankPageDetection flag was set to true in the job request.

11.3 Data Format

Pull Scan Job uses the ScanSettings (see ScanSettings section) in XML format. All the ScanSettings rules apply: all parameters are optional; the scanner determines the missing parameters based on:

- the Intent parameter (if provided)
- the scan capabilities default values
- internal sensors if available
- extra parameters are ignored if conflicting with the ‘Intent’ or each other.

Note: ScanDestinations SHOULD be omitted. It is a provision to extend the protocol to different transactional models.

11.4 Interface

Scan Jobs		
/{root}/ScanJobs		
POST	Description	Create a scan job

	Payload	IN	XML: /scan:ScanSettings (without scan:ScanDestinations)
	Status Code	201 Created – Location: {JobUri}	
		301 Moved - The client is redirected to a secure connection.	
		400 Bad Request - Request not understood due to wrong syntax	
		401 Unauthorized - The client is challenged for access credentials.	
		409 Conflict – Invalid input settings that conflict the device state or device capabilities or other settings in the payload.	
		500 Internal Server Error - Unknown internal error.	
		503 Service Unavailable – Device cannot handle this request at present. Retry later.	
{JobUri}			
DELETE	Description	Cancel a scan job	
	Payload	IN	None
	Status Code	200 OK – Success	
		301 Moved - The client is redirected to a secure connection.	
		401 Unauthorized - The client is challenged for access credentials.	
		404 Not Found - Resource is not found.	
		410 Gone - The resource used to exist but is gone.	
		500 Internal Server Error - Unknown internal error.	
		503 Service Unavailable – Server busy. Retry later	
{JobUri}/NextDocument			
GET	Description	Upload the next scan document (text or photo), using the scan job settings.	
	Payload	IN	HTTP Header: “TE: chunked” - the client MUST support chunked encoding
		OUT	HTTP Header: “Content-Location: {DocumentURI}” – this URI uniquely identifies the uploaded document. It is opaque to the client.

			HTTP Header: “Accept-Ranges: {bytes|none}” – this header indicates whether the client can attempt to use the “Range” header on the Document URI provided in the Content-Location header. If the value is set to “none” or the entire header is missing, the client cannot use “Range”. This header is optional: low end scanner MAY not support this feature. However, high end scanners SHOULD support it. Payload: raw binary data. Data format according to the job settings: e.g. image/jpeg or application/pdf
	Status Code	200 OK – Success: complete payload transmitted	
		301 Moved - The client is redirected to a secure connection.	
		401 Unauthorized - The client is challenged for access credentials.	
		404 Not Found – no more page. The last page has already been transmitted.	
		410 Gone – the scan job doesn’t exist anymore. May want to check the status.	
		500 Internal Server Error - Unknown internal error.	
		503 Service Unavailable – the job is active, but the scanner can’t return the payload at the moment. Retry later.	
{JobUri}/ScanImageInfo			
GET	Description	Retrieve the scan image info for the most recent scan page retrieved.	
	Payload	IN	None
		OUT	XML: /scan:ScanImageInfo
	Status Code	200 OK – Success:	
		301 Moved - The client is redirected to a secure connection.	
		401 Unauthorized - The client is challenged for access credentials.	
		404 Not Found – ScanImageInfo is no more available for the recent scan page. The info has already been transmitted.	

		410 Gone – the scan job doesn’t exist anymore. May want to check the status.	
		500 Internal Server Error - Unknown internal error.	
		503 Service Unavailable – the job is active, but the scanner can’t return the buffer info at the moment. Retry later.	
{Document URI} - *optional*			
GET	Description	This URL uniquely identifies the document (text or photo). It is useful in the context of a job with multiple documents. It can also be used if a long document’s transmission is aborted before the whole document is sent. The client can invoke this URL with a “Range” header to resume the transmission starting from the last byte count previously received by the client.	
	Payload	IN	HTTP Header: “Range: bytes={first missing byte}-”
		OUT	Payload: raw binary data. Data format according to the job settings: e.g. image/jpeg or application/pdf
	Status Code	200 OK – Success: partial payload transmitted	
		301 Moved - The client is redirected to a secure connection.	
		401 Unauthorized - The client is challenged for access credentials.	
		404 Not Found – Shouldn’t happen. The Document URI never existed.	
		410 Gone – the document existed but is no longer available	
		416 Requested Range Not Satisfiable – the Range header in the request does not match the range of data the scanner has cached.	
		500 Internal Server Error - Unknown internal error.	
		503 Service Unavailable – Server busy. Retry later	

11.5 Usage Flow

```
The client creates a scan job: pull-scan mode

The client application wants to scan a 4x6 grayscale
picture.
```

➔

```
POST /eSCL/ScanJobs HTTP/1.1
Host: 192.168.0.100
```

	Content-Type: text/xml
	Content-Length: 637
	
	<?xml version="1.0" encoding="UTF-8"?>
	<scan:ScanSettings xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
	  <pwg:Version>2.6</pwg:Version>
	  <scan:Intent>Photo</scan:Intent>
	  <pwg:ScanRegions>
	    <pwg:ScanRegion>
	      <pwg:Height>1200</pwg:Height>
	
	  <pwg:ContentRegionUnits>escl:ThreeHundredthsOfInches</pwg:ContentRegionUnits>
	      <pwg:Width>1800</pwg:Width>
	      <pwg:XOffset>0</pwg:XOffset>
	      <pwg:YOffset>0</pwg:YOffset>
	    </pwg:ScanRegion>
	  </pwg:ScanRegions>
	  <pwg:InputSource>Platen</pwg:InputSource>
	  <scan:ColorMode>Grayscale8</scan:ColorMode>
	  <scan:BlankPageDetection>true</scan:BlankPageDetection>
	</scan:ScanSettings>

←	HTTP/1.1 201 Created Location: http://192.168.1.100/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf
The client retrieves the first document	
→	GET /eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf/NextDocument HTTP/1.1 Host: 192.168.1.100 TE: chunked
←	HTTP/1.1 200 OK

	Content-Type: image/jpeg Transfer-Encoding: chunked Content-Location: /eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf/photo-1 Accept-Ranges: bytes ... JPEG binary data ... # sent in chunks <cr><lf>0<cr><lf><cr><lf> #last chunk
The client can retrieve the ScanImageInfo to know the length of the actual scanned image.	
→	GET /eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf/ScanImageInfo HTTP/1.1 Host: 192.168.1.100
←	HTTP/1.1 200 OK Content-Type: text/xml Content-Length: 1158 <?xml version="1.0" encoding="UTF-8"?> <scan:ScanImageInfo xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <pwg:JobUri>/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUri> <pwg:JobUuid>893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUuid> <scan:ActualWidth>2000</scan:ActualWidth> <scan:ActualHeight>3000</scan:ActualHeight> <scan:ActualBytesPerLine>2000</scan:ActualBytesPerLine> </scan:ScanImageInfo>
The client tries to retrieve more pages	

➔

```
GET /eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf/NextDocument HTTP/1.1
Host: 192.168.1.100
TE: chunked
```

🡐

```
HTTP/1.1 404 Not Found
```

It seems that we reached the last page. The client checks the status for confirmation.

➔

```
GET /eSCL/ScannerStatus HTTP/1.1
Host: 192.168.0.100
```

🡐

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 1158

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScannerStatus xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <pwg:State>Idle</pwg:State>
  <scan:Jobs>
   <scan:JobInfo>
     <pwg:JobUri>/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUri>
     <pwg:JobUuid>893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUuid>
     <scan:Age>40</scan:Age>
     <pwg:ImagesCompleted>1</pwg:ImagesCompleted>
     <pwg:JobState>Completed</pwg:JobState>
     <pwg:JobStateReasons>
       <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobStateReason>
     </pwg:JobStateReasons>
   </scan:JobInfo>
   <scan:JobInfo>
     <pwg:JobUri>/eSCL/ScanJobs/898d6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUri>
     <pwg:JobUuid>898d6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUuid>
     <scan:Age>220</scan:Age>
     <pwg:ImagesCompleted>5</pwg:ImagesCompleted>
     <pwg:JobState>Completed</pwg:JobState>
     <pwg:JobStateReasons>

   <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobStateReason>
     </pwg:JobStateReasons>
   </scan:JobInfo>
   </scan:Jobs>
</scan:ScannerStatus>
```

Multipage PDF document

```
➔ POST /eSCL/ScanJobs HTTP/1.1
Host: 192.168.0.100
Content-Type: text/xml
Content-Length: 860

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScanSettings xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <scan:Intent>Document</scan:Intent>
  <pwg:ScanRegions>
   <pwg:ScanRegion>
    <pwg:Height>3300</pwg:Height>
```

	<pwg:ContentRegionUnits>escl:ThreeHundredthsOfInches</pwg:ContentRegionUnits> <pwg:Width>2400</pwg:Width> <pwg:XOffset>0</pwg:XOffset> <pwg:YOffset>0</pwg:YOffset> </pwg:ScanRegion> </pwg:ScanRegions> <!—FROM eSCL 2.0 clients → <pwg:DocumentFormat>application/pdf</pwg:DocumentFormat> <!—FROM eSCL 2.1 clients → <scan:DocumentFormatExt>application/pdf</scan:DocumentFormatExt> <pwg:InputSource>Feeder</pwg:InputSource> <scan:ColorMode>RGB24</scan:ColorMode> <scan:Duplex>true</scan:Duplex> </scan:ScanSettings>
←	HTTP/1.1 201 Created Location: http://192.168.1.100/eSCL/ScanJobs/898e6fcd-487f-4056-a8c9-a87709b85daf
The client retrieves the PDF document (with multiple pages)	
→	GET /eSCL/ScanJobs/898e6fcd-487f-4056-a8c9-a87709b85daf/NextDocument HTTP/1.1 Host: 192.168.1.100 TE: chunked
←	HTTP/1.1 200 OK Content-Type: application/pdf Transfer-Encoding: chunked Content-Location: /eSCL/ScanJobs/898e6fcd-487f-4056-a8c9-a87709b85daf/doc-10 #remember: this is opaque but unique Accept-Ranges: bytes

	... PDF binary data ... # many pages (325496 bytes) of the PDF document XXXXX Lost connection – end of document missing
The client keeps the first 325496 bytes of the document, and attempts to resume the transaction for the end of the document	
→	GET /eSCL/ScanJobs/898e6fcd-487f-4056-a8c9-a87709b85daf/doc-10 HTTP/1.1 Host: 192.168.1.100 TE: chunked Range: bytes=325496-
←	HTTP/1.1 200 OK Content-Type: application/pdf Transfer-Encoding: chunked Accept-Ranges: bytes ... PDF binary data ... # last fragment of the PDF document, sent in chunks <cr><lf>0<cr><lf><cr><lf> #last chunk
The client checks if there is a new document	
→	GET /eSCL/ScanJobs/898e6fcd-487f-4056-a8c9-a87709b85daf/NextDocument HTTP/1.1 Host: 192.168.1.100 TE: chunked
←	HTTP/1.1 404 Not Found
It seems that we reached the last document. The client checks the status for confirmation.	
→	GET /eSCL/ScannerStatus HTTP/1.1 Host: 192.168.0.100
←	HTTP/1.1 200 OK Content-Type: text/xml

```
Content-Length: 1158


<?xml version="1.0" encoding="UTF-8"?>
<scan:ScannerStatus
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03
" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <pwg:State>Idle</pwg:State>
  <scan:Jobs>
   <scan:JobInfo>
     <pwg:JobUri>/eSCL/ScanJobs/898e6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUri>
     <pwg:JobUuid>898e6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUuid>
     <scan:Age>40</scan:Age>
     <pwg:ImagesCompleted>2</pwg:ImagesCompleted>
     <pwg:JobState>Completed</pwg:JobState>
     <pwg:JobStateReasons>

  <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobSt
ateReason>
     </pwg:JobStateReasons>
   </scan:JobInfo>
   <scan:JobInfo>
     <pwg:JobUri>/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUri>
     <pwg:JobUuid>893e6fcd-487f-4056-a8c9-
a87709b85daf</pwg:JobUuid>
     <scan:Age>200</scan:Age>
     <pwg:ImagesCompleted>2</pwg:ImagesCompleted>
     <pwg:JobState>Completed</pwg:JobState>
     <pwg:JobStateReasons>

  <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobSt
ateReason>
```

```
    </pwg:JobStateReasons>
   </scan:JobInfo>
  </scan:Jobs>
</scan:ScannerStatus>
```

12 Cancel a Scan Job

Cancelling the job is performed by DELETEing the job instance URL

The client cancels an active scan job
DELETE /eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf HTTP/1.1 Host: 192.168.1.100
HTTP/1.1 200 OK

13 Stored Job Requests

A user can reserve a scan job by requesting the device to hold on to the job request until the user confirms at the device front panel. User can also reserve the scan job with an optional secure PIN.

Clients can track the stored job requests through the ScannerStatus. When the scan job is put on hold, the ScannerStatus SHOULD report the job state be in the PendingHeld state and the JobStateReason as JobHeldByService.

If user does not initiate the job at the front panel within the specific timeout mentioned in the ScannerCapabilities, the device MUST remove the job request from its stored jobs list. Alternatively, the client can cancel the job request by sending a DELETE request to the JobURI.

13.1 Usage Flow

The client creates a reserved scan job request. The client application wants to start a scan job with the default scan settings.

→

```
POST /eSCL/ScanJobs HTTP/1.1
Host: 192.168.0.100
Content-Type: text/xml
Content-Length: 637

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScanSettings
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03"
xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
  <pwg:Version>2.6</pwg:Version>
  <scan:StoredJobRequest>
    <scan:JobName>myHostName</scan:JobName>
    <scan:PIN>q1w2e3r4</scan:PIN>
  </scan:StoredJobRequest>
</scan:ScanSettings>
```

←

```
HTTP/1.1 201 Created
Location: http://192.168.1.100/eSCL/ScanJobs/893e6fcd-487f-
4056-a8c9-a87709b85daf
```

If the client gets the current state of this job request, it should be in PendingHeld state.

→

```
GET /eSCL/ScannerStatus HTTP/1.1
Host: 192.168.0.100
```

←

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 1158

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScannerStatus xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
      <pwg:Version>2.6</pwg:Version>
      <pwg:State>Idle</pwg:State>
      <scan:Jobs>
        <scan:JobInfo>
          <pwg:JobUri>/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUri>
          <pwg:JobUuid>893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUuid>
          <scan:Age>10</scan:Age>
          <pwg:ImagesCompleted>0</pwg:ImagesCompleted>
          <pwg:ImagesToTransfer>0</pwg:ImagesToTransfer>
          <pwg:JobState>PendingHeld</pwg:JobState>
          <pwg:JobStateReasons>
             <pwg:JobStateReason>JobHeldByService</pwg:JobStateReason>
          </pwg:JobStateReasons>
        </scan:JobInfo>
      </scan:Jobs>
</scan:ScannerStatus>
```

User goes to the device front panel, retrieves the reserved job from an existing list of jobs and keys in the secret PIN. Device verifies that the PIN entered by user matches with the one in the scan payload. Then the scanner starts the actual job.

Client can start pulling the scan image once the JobState moves to Processing.

→

```
GET /eSCL/ScannerStatus HTTP/1.1
Host: 192.168.0.100
```

←

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 1158

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScannerStatus xmlns:scan="http://schemas.hp.com/imaging/e
scl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/s
m">
      <pwg:Version>2.6</pwg:Version>
      <pwg:State>Processing</pwg:State>
      <scan:Jobs>
          <scan:JobInfo>
               <pwg:JobUri>/ScanJobs/893e6fcd-487f-4056-a8c9-a8
7709b85daf</pwg:JobUri>
               <pwg:JobUuid>893e6fcd-487f-4056-a8c9-a87709b85da
f</pwg:JobUuid>
               <scan:Age>10</scan:Age>
               <pwg:ImagesCompleted>0</pwg:ImagesCompleted>
               <pwg:ImagesToTransfer>0</pwg:ImagesToTransfer>
               <pwg:JobState>Processing</pwg:JobState>
               <pwg:JobStateReasons>
                   <pwg:JobStateReason>JobScanning</pwg:JobStat
eReason>
               </pwg:JobStateReasons>
          </scan:JobInfo>
      </scan:Jobs>
</scan:ScannerStatus>
```

After the client pulls all the scan images, the JobState moves to Completed.

→

```
GET /eSCL/ScannerStatus HTTP/1.1
Host: 192.168.0.100
```

←

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 1158

<?xml version="1.0" encoding="UTF-8"?>
```

```xml
<scan:ScannerStatus xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
	<pwg:Version>2.6</pwg:Version>
	<pwg:State>Idle</pwg:State>
	<scan:Jobs>
		<scan:JobInfo>
			<pwg:JobUri>/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUri>
			<pwg:JobUuid>893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUuid>
			<scan:Age>10</scan:Age>
			<pwg:ImagesCompleted>0</pwg:ImagesCompleted>
			<pwg:ImagesToTransfer>0</pwg:ImagesToTransfer>
			<pwg:JobState>Completed</pwg:JobState>
			<pwg:JobStateReasons>
				<pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobStateReason>
			</pwg:JobStateReasons>
		</scan:JobInfo>
	</scan:Jobs>

</scan:ScannerStatus>
```

14 Security

eSCL is based on HTTP, and eSCL transactions can be performed over:

- HTTP – port 80: HTTP
- HTTPS – port 443: HTTP over TLS
- HTTP Upgrade – port 80: either client or server can request the connection to be upgraded to TLS using the HTTP Upgrade mechanism [7]
- External secure transport: for example, XMPP

From eSCL 2.97 onwards, scanners with user management capabilities SHOULD support at least one authentication mechanism (Basic, Digest, TLS Certificate, OAuth 2.0). A scanner SHOULD only challenge a client for authentication over a TLS connection.

The scanner administrator MAY configure security settings, including which specific authentication methods are active and which versions of TLS are enabled. If TLS is active, both clients and scanners MUST support TLS 1.3 [8] or later. A scanner MAY support TLS 1.1 but it MUST NOT be enabled by default.

~~The scanner administrator MAY configure some settings to force the eSCL clients to use either SSL/TLS transport or authentication or both. If secure transport scanning is enforced, the scanner MUST redirect the client application to use HTTPS if the request comes over HTTP. If authentication is enforced, the scanner MUST verify the client with any of the authentication mechanisms: Basic auth, Digest auth, Client certificate check, etc.,~~

~~It is highly recommended that the admin configures to perform HTTP Basic authentication only over a SSL/TLS connection. However, Digest Authentication does not need SSL/TLS connection.~~

These admin settings SHOULD not affect the requests over XMPP channel, as XMPP channel is considered to be already secure. USB requests are secure with respect to the transport, so USB requests MUST NOT be redirected to HTTPS even if the admin settings require that, but USB requests SHOULD support authentication if admin setting enforces authentication.

When the client application is redirected, it MUST use the same hostname and same relative URL. The scanner MUST enforce this requirement in the Location header (requirement for some JavaScript environments).

Below are the job specific resources that MUST adhere to the security requirements if admin makes it mandatory.

ResourceType	ResourceURI
ScannerStatus	/{root}/ScannerStatus

ResourceType	ResourceURI
ScanJobs	/{root}/ScanJobs
ScanJob	/{root}/ScanJobs/{job-id}
ScanImageInfo	/{root}/ScanJobs/{job-id}/ScanImageInfo
ScanData	URL is dynamic, specified at job creation

```
➔ POST /eSCL/ScanJobs HTTP/1.1
  Host: 192.168.0.100
  Content-Type: text/xml
  Content-Length: 255

  <?xml version="1.0" encoding="UTF-8"?>
  <scan:ScanSettings
  xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03"
  xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
    <pwg:Version>2.6</pwg:Version>
    <scan:Intent>Photo</scan:Intent>
  </scan:ScanSettings>
```

```
← HTTP/1.1 301 Moved Permanently
  Location: https://192.168.0.100/eSCL/ScanJobs
```

```
Reconnect with SSL at 192.168.0.100 port 443
```

```
➔ POST /eSCL/ScanJobs HTTP/1.1
  Host: 192.168.0.100:443
  Content-Type: text/xml
  Content-Length: 255

  <?xml version="1.0" encoding="UTF-8"?>
  <scan:ScanSettings
  xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03"
  xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
```

	<pwg:Version>2.6</pwg:Version> <scan:Intent>Photo</scan:Intent> </scan:ScanSettings>
←	HTTP/1.1 401 Unauthorized WWW-Authenticate: Basic realm="scannerAdmin"
→	POST /eSCL/ScanJobs HTTP/1.1 Host: 192.168.0.100:443 Authorization: Basic TGF1cmVudFBpem90V3JvdGVUaGlzIQ== Content-Type: text/xml Content-Length: 255 <?xml version="1.0" encoding="UTF-8"?> <scan:ScanSettings xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm"> <pwg:Version>2.6</pwg:Version> <scan:Intent>Photo</scan:Intent> </scan:ScanSettings>
←	HTTP/1.1 201 Created Location: https://192.168.0.100:443/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf
Get the scanned document Close the SSL connection	

15 Scan Job (Push)

An eSCL scanner MAY support the “push” model in addition to the “pull” model.

Scan jobs is divided in two major use cases:

- Pull Scan – the client application paces the retrieval of the scan pages
- Push Scan – the scanner sends the image back to the client or a distinct server: the scanner is uploading scan images to a ‘destination’ specified in the scan request.

The scanner detects the Pull or Push mode based on the absence or the presence (respectively) of the ScanDestinations element in ScanSettings. In other words, the absence of the ScanDestinations field is the only indication that the client wants to initiate a Pull scan job.

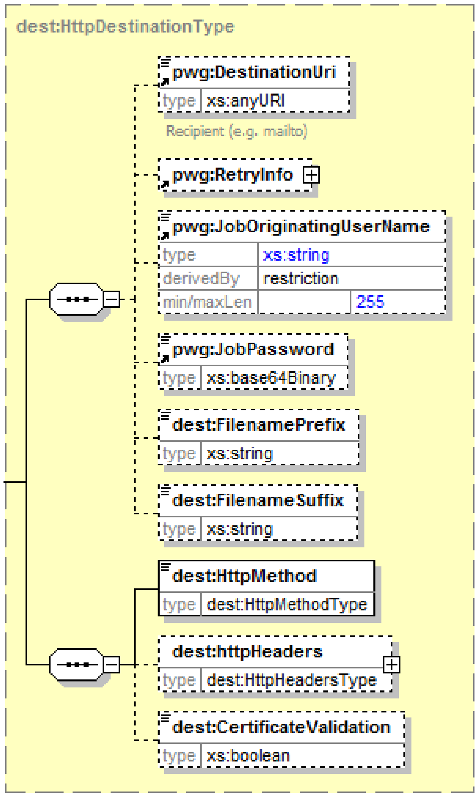
This section covers only the push mechanism. The client uses the push scan mode to avoid firewall issues. Moreover, it allows the scanner to connect to 3rd party web services directly and provides specific credentials to access protected user accounts. To instantiate a push-scan job, the ScanSettings are identical to that of a pull-scan job, except for the ScanDestination parameter.

15.1 Scan Destinations

The scan destinations are specific for Push-scan jobs. They provide the necessary information to upload a document to a web service, like a cloud content repository.

The targeted server MUST support chunked encoding.

The ScanDestinations can provide a full description of the HTTP Request parameters required to upload the scan output data (a.k.a. HttpDestination namespace=”http://www.hp.com/imaging/httpdestinations/2011/10/13”).



Scan images can be uploaded to multiple destinations if the client requests so. Even if there is failure in uploading the images to certain destinations, the scanner MUST continue uploading the images to all the other destinations. The status of each of the destination uploads can be tracked with the ScannerStatus.

15.1.1.1 HttpDestination

- DestinationUri: This is the root URL location, where the client expects the scanner to upload the scanned document.
- DestinationReferenceID:. A UUID or some unique ID that client assigns to this destination URI. From eSCL 2.6 onwards,a client SHOULD send this ID in the scan job request if it wants to track the status of the image uploads to the destinations. The scanner MUST mention the ReferenceID and not the DestinationUri in the ScannerStatus xml. If this ID is not sent by the client in the input settings, the scanner MUST NOT send the DestinationStatuses element in the ScannerStatus xml.
- RetryInfo: indicates the number of times the scanner should retry uploading in case of problems, the time interval between retries, and how long to wait before timing out.
- JobOriginatingUserName and JobPassword: username and password to use when challenged by the server for HTTP authentication (e.g. Basic or Digest)
- Filename Prefix and Suffix: when specified, the scanner will extend the DestinationUri with "/{FilenamePrefix}{incrementing integer}{FilenameSuffix}". If both fields are missing, the DestinationUri is used as is, even if multiple documents are uploaded. For example, if prefix is 'img' and suffix is '.jpg' and if the scanner uploads 10 images, the upload URLs would be: {DestinationUri}/img1.jpg then {DestinationUri}/img2.jpg until {DestinationUri}/img10.jpg. If neither the prefix nor the suffix are provided (or are empty), all 10 images are uploaded to the single URL provided in DestinationUri.
- HttpMethod: indicates what HTTP verb to use for the request: usually POST or PUT
- HttpHeaders: these optional headers can be specified to provide security credentials to access the storage service. The content must be XML character safe: <>"' replaced by XML equivalent sequences.
- CertificateValidation: indicates whether the scanner must use its trusted store's Certificate Authority to authenticate the server. If scanner can store the CA certificate, but if the certificate is not installed on the scanner, then scanner SHOULD not start the scan job and SHOULD report a 409 HTTP response.

If the URL and the HTTP headers are confidential, the client must request the scanner to use TLS/SSL before sending the HTTP request: i.e. the URL in {dest:DestinationUri} must use the HTTPS scheme https://...

15.2 Scanner Capabilities

- ScanToDestinationsSupport: This complex type is used to report the PUSH scan specific capabilities.
    - MaxDestinations: Maximum number of destinations that the scanned images can be pushed to. Supporting more than one destination MAY need the scanner to store all the scanned images in its memory.
    - CertificateValidationSupport: Indicates if the scanner can store a trusted Certificate Authority certificate and use it to validate the remote servers' certificate.
    - UriSchemes: List of URI schemes supported by the device for PUSH scans. PUSH scan devices MUST support 'http' and 'https' URI schems. Devices MAY support other file transfer URI schemes.
    - PreConfiguredDestinations: List of pre-configured destinations on the device to help the user in choosing a destination. User MAY choose among the pre-configured destinations or specify a different destination.

15.3 Job State

With Push scan, some job states provide additional information:

- Aborted: end state – either a communication error (the scanner couldn't connect to the server for push-scan job) or a security error (when connecting to the server: e.g. server certificate validation failed)

15.4 Scanner Status

For Push scans, the job info SHOULD provide additional info. The DestinationStatuses gives the current state of the push scan job to a destination.

- DestinationReferenceID: The reference ID that the client has assigned to the destination. This is used to track the status of the image uploads to a destination. If this ID was not sent by the client while initiating a scan job, the scanner MUST NOT send the DestinationStatuses element in the ScannerStatus xml.
- ImagesTransferred: Number of images or pages pushed to the destination up to the time of the status request.
- ImagesToTransfer: The number of images or pages that are yet to be transferred.
- TransferRetryCount: The number of consecutive transmission retries that have been attempted for the current or the most recent transmission operation for this job. 0 means no retries attempted for the current transmission operation.

- TotalRetryCount: The cumulative number of transmission retries that have been attempted for this destination, which includes the retries for all the scan pages of this job.
- TransferState, TransferStateReason: The current state and reason for the scan image transfer to the destination.

15.5 Usage Flow

A typical scan job request would be as follows:

The client sends a request to start a push scan job and send the scanned image to multiple HTTP destinations

➔

```
POST /eSCL/ScanJobs HTTP/1.1
Host: 192.168.0.100
Content-Type: text/xml
Content-Length: 1635

<?xml version="1.0" encoding="UTF-8"?>
<scan:ScanSettings
xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03
" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm"
xmlns:dest="http://schemas.hp.com/imaging/httpdestination
/2011/10/13">
  <pwg:Version>2.6</pwg:Version>
  <scan:Intent>Photo</scan:Intent>
  <pwg:ScanRegions>
   <pwg:ScanRegion>
     <pwg:Height>1200</pwg:Height>

  <pwg:ContentRegionUnits>escl:ThreeHundredthsOfInches</p
wg:ContentRegionUnits>
     <pwg:Width>1800</pwg:Width>
     <pwg:XOffset>0</pwg:XOffset>
     <pwg:YOffset>0</pwg:YOffset>
   </pwg:ScanRegion>
  </pwg:ScanRegions>
  <pwg:InputSource>Platen</pwg:InputSource>
  <scan:ColorMode>Grayscale8</scan:ColorMode>
  <!--  note that the contents of Destination elements
are strings
       with XML safe characters.
   For example, the quote character (“) is replaced by
&quot;
   -->
  <scan:ScanDestinations>
   <dest:HttpDestination>

  <pwg:DestinationUri>https://storage.com/accounts/123456
789</pwg:DestinationUri>
  <dest:ReferenceID>12345qwert</dest:ReferenceID>
  <pwg:RetryInfo>
      <pwg:NumberOfRetries>3</pwg:NumberOfRetries>
      <pwg:RetryInterval>60</pwg:RetryInterval>
      <pwg:RetryTimeOut>60</pwg:RetryTimeOut>
     </pwg:RetryInfo>

  <pwg:JobOriginatingUserName>JohnSmith</pwg:JobOriginati
ngUserName>

  <pwg:JobPassword>TGF1cmVudFBpem90V3JvdGVUaGlzIQ==</pwg:
JobPassword>
     <dest:HttpMethod>POST</dest:HttpMethod>
     <dest:httpHeaders>
      <dest:HttpHeader>Cookie: token=156724
domain=&quot;my.storage.com&quot;</dest:HttpHeader>
     </dest:httpHeaders>

  <dest:CertificateValidation>true</dest:CertificateValid
ation>
   </dest:HttpDestination>
   <dest:HttpDestination>
<pwg:DestinationUri>https://docs.google.com/accounts/1234
</pwg:DestinationUri>
```

	<dest:ReferenceID>a1s2d3f4g5</dest:ReferenceID>
	<pwg:JobOriginatingUserName>JohnSmith</pwg:JobOriginatingUserName>
	<pwg:JobPassword>TGF1cmVudFBpem90V3JvdGVUaGlzIQ==</pwg:JobPassword>
	    <dest:HttpMethod>POST</dest:HttpMethod>
	  <dest:CertificateValidation>true</dest:CertificateValidation>
	   </dest:HttpDestination>
	  </scan:ScanDestinations>
	</scan:ScanSettings>

←	HTTP/1.1 201 Created Location: http://192.168.1.100/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf
The scanner connects to my.storage.com using SSL (https://) and sends the HTTP request with the scanned image.	
←	POST /accounts/123456789 HTTP/1.1 Host: my.storage.com:443 Authentication: Basic TGF1cmVudFBpem90V3JvdGVUaGlzIQ== Cookie: token=156724 domain="my.storage.com" Content-Type: application/pdf Content-Length: 2764712 ... PDF binary data ...
→	HTTP/1.1 200 OK
The scanner connects to docs.google.com using SSL (https://) and sends the scanned image.	
←	POST /accounts/1234 HTTP/1.1 Host: docs.google.com:443 Authentication: Basic TGF1cmVudFBpem90V3JvdGVUaGlzIQ== Cookie: token=156724 domain="docs.google.com" Content-Type: application/pdf Content-Length: 2764712

	... PDF binary data ...
→	HTTP/1.1 200 OK
Meanwhile the client queries to scanner to track the execution of the job Until job …/893e6fcd-487f-4056-a8c9-a87709b85dafhas the state of Completed, CanceledbyDevice, CommunicationError or TransactionError	
→	GET /eSCL/ScannerStatus HTTP/1.1 Host: 192.168.0.100

←	HTTP/1.1 200 OK
	Content-Type: text/xml
	Content-Length: 945
	
	<?xml version="1.0" encoding="UTF-8"?>
	<scan:ScannerStatus xmlns:scan="http://schemas.hp.com/imaging/escl/2011/05/03" xmlns:pwg="http://www.pwg.org/schemas/2010/12/sm">
	  <pwg:Version>2.6</pwg:Version>
	  <pwg:State>Idle</pwg:State>
	  <scan:Jobs>
	   <scan:JobInfo>
	    <pwg:JobUri>/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUri>
	    <pwg:JobUuid>893e6fcd-487f-4056-a8c9-a87709b85daf</pwg:JobUuid>
	    <scan:Age>10</scan:Age>
	    <pwg:ImagesCompleted>1</pwg:ImagesCompleted>
	    <pwg:ImagesToTransfer>0</pwg:ImagesToTransfer>
	    <pwg:JobState>Completed</pwg:JobState>
	    <pwg:JobStateReasons>
	
	  <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobStateReason>

```
    </pwg:JobStateReasons>
   </scan:JobInfo>
   <scan:JobInfo>
    <pwg:JobUri>/eSCL/ScanJobs/893e6fcd-487f-4056-a8c9-
a87709b845af</pwg:JobUri>
    <pwg:JobUuid>893e6fcd-487f-4056-a8c9-
a87709b845af</pwg:JobUuid>
    <scan:Age>220</scan:Age>
    <pwg:ImagesCompleted>1</pwg:ImagesCompleted>
    <pwg:ImagesToTransfer>0</pwg:ImagesToTransfer>
    <pwg:JobState>Completed</pwg:JobState>
    <pwg:JobStateReasons>

  <pwg:JobStateReason>JobCompletedSuccessfully</pwg:JobSt
ateReason>
    </pwg:JobStateReasons>
     <scan:DestinationStatuses>
     <scan:DestinationStatus>


<scan:DestinationReferenceID>12345qwert<scan:DestinationR
eferenceID>
      <scan:ImagesTransferred>0</scan:ImagesTransferred>
      <scan:ImagesToTransfer>1</scan:ImagesToTransfer>
  <scan:TransferRetryCount>0</scan:TransferRetryCount>
      <scan:TotalRetryCount>0</scan:TotalRetryCount>
      <scan:TransferState>Completed</scan:TransferState>
  <scan:TransferStateReason>JobCompletedSuccessfully</sca
n:TransferStateReason>
     </scan:DestinationStatus>
     <scan:DestinationStatus>
<scan:DestinationReferenceID>a1s2d3f4g5<scan:DestinationR
eferenceID>
  <scan:ImagesTransferred>1</scan:ImagesTransferred>
      <scan:ImagesToTransfer>0</scan:ImagesToTransfer>
  <scan:TransferRetryCount>0</scan:TransferRetryCount>
```

```
         <scan:TotalRetryCount>0</scan:TotalRetryCount>
         <scan:TransferState>Completed</scan:TransferState>
  <scan:TransferStateReason>JobCompletedSuccessfully</sca
n:TransferStateReason>
        </scan:DestinationStatus>
      </scan:DestinationStatuses>         </scan:JobInfo>
  </scan:Jobs>
</scan:ScannerStatus>
```

The Push-Scan transaction can be secured by one or more of the following methods:

- specifying https:// in the Upload URL field
- using a Fat URL to authenticate the scanner
- specifying extra HTTP headers for authentication sake
- specifying Username and Password to use when challenged for HTTP authentication