How to Use the Internet Printing Protocol

v2018.0918

Michael R Sweet, Peter Zehler

Table of Contents

Chapter 1: Introduction

What is IPP?

The Internet Printing Protocol ("IPP") is a secure application level protocol used for network printing. The protocol allows a Client to ask a Printer about its capabilities and defaults (supported media sizes, two-sided printing, etc.), and the state of the Printer (paper out/jam, low ink/toner, etc.) and any Print Jobs. The Client can also submit files for printing and subsequently cancel them. IPP is supported by all modern network printers and supercedes all legacy network protocols including port 9100 printing and LPD/lpr.

IPP is widely implemented in software as well, including the following open source projects:

- C-based: CUPS and PWG IPP Sample Code
- Java: Java IPP Client Implementation and Core parser for a Java implementation of IPP
- Javascript IPP Client Implementation
- Python IPP Client Implementation

IPP Overview

IPP defines an abstract model for printing, including operations with common semantics (business logic) for working with the model's objects. Because the same semantics of IPP are followed by all printers, the client (software) does not need to know the internal details of the printer (hardware).

IPP uses HTTP as its transport protocol. Each IPP request is a HTTP POST with a binary IPP message and print file, if any, in the request message body. The corresponding IPP response is returned in the POST response. HTTP connections can be unencrypted, upgraded to TLS encryption using an HTTP OPTIONS request, or encrypted immediately (HTTPS). HTTP POST requests can also be authenticated using any of the usual HTTP mechanisms.

> Note: Legacy network protocols do not support authentication, authorization, or privacy (encryption).

Printers are identified using Universal Resource Identifiers ("URIs") with the "ipp" or "ipps" scheme. Print jobs are identified using the printer's URI and a job number that is unique to that printer. The following are example printer URIs:

```
ipp://printer.example.com/ipp/print
ipps://printer2.example.com:443/ipp/print
ipps://server.example.com/ipp/print/printer3
```

These are mapped to "http" and "https" URLs, with a default port number of 631 for IPP. For example, the previous IPP URIs would be mapped to:

```
http://printer.example.com:631/ipp/print
https://printer2.example.com/ipp/print
https://server.example.com:631/ipp/print/printer3
```

> Note: The resource path "/ipp/print" is commonly used by IPP printers, however there is no hard requirement to follow that convention and older IPP printers used a variety of different locations. Consult your printer documentation or the printer's Bonjour registration information to determine the proper hostname, port number, and path to use for your printer.

IPP Operations

The following IPP operations are commonly used:

- Create-Job: Create a new (empty) print job.
- Send-Document: Add a document to a print job.
- Print-Job: Create a new print job with a single document.
- Get-Printer-Attributes: Get printer status and capabilities.
- Get-Jobs: Get a list of queued jobs.
- Get-Job-Attributes: Get job status and options.
- Cancel-Job: Cancel a queued job.

The IANA IPP Registry lists all of the registered IPP operations.

> Note: IPP provides two ways to print a single file - using the Print-Job operation or using a combination of the Create-Job and Send-Document operations. Clients typically use the Create-Job and Send-Document operations so that the job can be more easily canceled while the document data is being transferred to the printer.

IPP Message Encoding

IPP messages use a common format for both requests (from the client to the printer) and responses (from the printer to the client). Each IPP message starts with a version number (2.0 is the most common), an operation (request) or status (response) code, a request number, and a list of attributes. Attributes are named and have strongly typed values such as:

- "collection": A list of key/value pairs that have been grouped together.
- "enum": Integer enumerated values, typically starting at 3, where each value has a specific meaning.

- "integer": 32-bit signed integers from -2147483648 to 2147483647.
- "keyword": A lowercase string identifier like "one-sided" or "iso_a4_210x297mm".
- "mimeMediaType": A MIME media type like "text/plain" or "application/pdf".
- "name": A human-readable name like "Bob Smith".
- "text": A human-readable string like "Printer is out of paper."
- "uri": A Universal Resource Identifier like "https://www.example.com" or "ipps://printer.example.com/ipp/print".

Attributes are also placed in groups according to their usage - the 'operation' group for attributes used for the operation request or response, the 'job' group for print job attributes, and so forth.

The first two attributes in an IPP message are always:

1. "attributes-charset" which defines the character set to use for all name and text strings, and
2. "attributes-natural-language" which defines the default language for those strings, e.g., "en" for English, "fr" for French, "ja" for Japanese, etc.

The next attributes must be the printer's URI ("printer-uri") and, if the request is targeting a print job, the job's ID number ("job-id").

Most requests also include the "requesting-user-name" attribute that provides the name of the user.

A request containing an attached print file includes the MIME media type for the file ("document-format"). The media type is 'text/plain' for text files, 'image/jpeg' for JPEG files, 'application/pdf' for PDF files, etc.

The following example encodes a Print-Job request using the ipptool test file format:

```
{
    VERSION 2.0
    OPERATION Print-Job
    REQUEST-ID 42

    GROUP operation-attributes-tag
    ATTR charset "attributes-charset" "utf-8"
    ATTR naturalLanguage "attributes-natural-language" "en"
    ATTR uri "printer-uri" "ipp://printer.example.com/ipp/print"
    ATTR name "requesting-user-name" "John Doe"
    ATTR mimeMediaType "document-format" "text/plain"

    FILE "testfile.txt"
}
```

The same request using the CUPS API would look like the following:

```
#include <cups/cups.h>

...

http_t *http;
ipp_t *request, *response;

http = httpConnect2("printer.example.com", 631, NULL, AF_UNSPEC, HTTP_ENCRYPTION_

request = ippNewRequest(IPP_OP_PRINT_JOB);
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI, "printer-uri", NULL, "ipp:/
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME, "requesting-user-name", NU
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_MIMETYPE, "document-format", NUL

response = cupsDoFileRequest(http, request, "/ipp/print", "testfile.txt");

ipp_attribute_t *attr;
const char *name;
char value[2048];

for (attr = ippFirstAttribute(response); attr; attr = ippNextAttribute(response))
{
  name = ippGetName(attr);

  if (name)
  {
    ippAttributeString(attr, value, sizeof(value));
    printf("%s=%s\n", name, value);
  }
}
```

And this is how you'd send a Print-Job request using the nodejs API:

```
var ipp = require("ipp");
var printer = ipp.Printer("http://printer.example.com:631/ipp/print");
var fs = require("fs");
var document;

fs.readFile("testfile.txt", function(err, data) {
  if (err) throw err;

  document = data;
});

var msg = {
  "operation-attributes-tag": {
    "requesting-user-name": "John Doe",
    "document-format": "text/plain"
  },
  data: document;
};

printer.execute("Print-Job", msg, function(err, res) {
```

```
        console.log(err);
        console.log(res);
});
```

The response message uses the same version number, request number, character set, and natural language values as the request. A status code replaces the operation code in the initial message header - for the Print-Job operation the printer will return the 'successful-ok' status code if the print request is successful or 'server-error-printer-busy' if the printer is busy and wants you to try again at a later time.

The character set and natural language values in the response are followed by operation-specific attributes. For example, the Print-Job operation returns the print job identifier ("job-id") and state ("job-state" and "job-state-reasons") attributes.

> You can learn more about the IPP message encoding by reading the Internet Printing Protocol/1.1: Encoding and Transport document.

Chapter 2: Printers

What are Printers?

Printers in IPP are objects that represent real or virtual (for saving, emailing, etc.) output devices. Printer objects provide attributes that describe the status of the printer (printing something, out of paper, etc.), the capabilities of the printer (what paper sizes are supported, can the printer reproduce color, can the printer staple the output, etc.), and general information about the printer (where the printer is located, the URL for the printer's administrative web page, etc.) Printers also manage a queue of print jobs.

Printer Status Attributes

Printers provide two main status attributes: "printer-state" and "printer-state-reasons". The "printer-state" attribute is a number that describes the general state of the printer:

- '3': The printer is idle.
- '4': The printer is processing a print job.
- '5': The printer is stopped and requires attention.

The "printer-state-reasons" attribute is a list of strings that provide details about the printer's state:

- 'none': Everything is super, nothing to report.
- 'media-needed': The printer needs paper loaded.
- 'toner-low': The printer is low on toner.
- 'toner-empty': The printer is out of toner.
- 'marker-supply-low': The printer is low on ink.
- 'marker-supply-empty': The printer is out of ink.

The string may also have a severity suffix ("-error", "-warning", or "-report") to tell the clients whether the reason affects the printing of a job.

> Note: The IANA IPP registry lists all of the registered strings for the "printer-state-reasons" attribute. All strings are in English but can be localized using message catalogs provided by each printer.

Many printers also provide status attributes for alerts ("printer-alert"), consumables ("printer-supply", "printer-supply-description", and "printer-supply-info-uri"), input trays ("printer-input-tray"), output trays ("printer-output-tray"), and so forth.

Printer Capability Attributes

Printers provide many capability attributes, including:

- "ipp-features-supported": A list of IPP features that are supported, for example 'ipp-everywhere' and 'icc-color-matching'.
- "ipp-versions-supported": A list of IPP versions that are supported, for example '1.1' and '2.0'.
- "operations-supported": A list of IPP operations that are supported, for example Print-Job, Create-Job, Send-Document, Cancel-Job, Get-Jobs, Get-Job-Attributes, and Get-Printer-Attributes.
- "charset-supported": A list of character sets that are supported ('utf-8' is required.)
- "job-creation-attributes-supported": A list of IPP attributes that are supported when submitting print jobs, for example 'media', 'media-col', and 'print-quality'.
- "document-format-supported": A list of file formats that can be printed, for example 'application/pdf' and 'image/pwg-raster'.
- "media-supported" and "media-col-database": A list of paper sizes and types that are supported, for example 'na_letter_8.5x11in' and 'iso_a4_210x297mm'.
- "media-ready" and "media-col-ready": A list of paper sizes and types that are loaded, for example 'na_letter_8.5x11in' and 'iso_a4_210x297mm'.
- "copies-supported": The maximum number of copies that can be produced, for example '99'.
- "sides-supported": A list of supported one and two sided printing modes, for example 'one-sided', 'two-sided-long-edge', and 'two-sided-short-edge'.
- "print-quality-supported": A list of supported print qualities, for example '3' (draft), '4' (normal), and '5' (high).
- "print-color-mode-supported": A list of supported color printing modes, for example 'bi-level', 'monochrome', and 'color'.
- "print-scaling-supported": A list of supported scaling modes, for example 'auto', 'fill', and 'fit'.
- "printer-resolution-supported": A list of supported print resolutions, for example '300dpi' and '600dpi'.
- "page-ranges-supported": Specifies whether page ranges are supported (boolean).
- "finishings-supported" and "finishings-col-database": A list of finishing processes (staple, punch, fold, etc.) that are supported.
- "finishings-ready" and "finishings-col-ready": A list of finishing processes that can be requested without stopping the printer.

Printer Information Attributes

Printers provide seven main description attributes: "printer-uri-supported", "uri-authentication-supported", "uri-security-supported", "printer-info",

"printer-more-info", "printer-location", and "printer-geo-location".

The "printer-uri-supported" attribute lists the supported printer URI values. The "uri-authentication-supported" attribute lists the authorization and access control requirements for each of the supported printer URI values. Similarly, the "uri-security-supported" attribute lists the encryption requirements for each of the supported printer URI values.

The "printer-info" attribute provides a textual description of the printer and often defaults to the make and model of the printer. The "printer-more-info" attribute provides a URL to the printer's administrative web page.

The "printer-location" attribute provides a textual location of the printer, for example 'Second floor near the break room.'. The "printer-geo-location" attribute provides the geographic location of the printer, if known, as a geo: URI.

Querying the Printer Attributes

The Get-Printer-Attributes operation is used to query any of the printer attributes mentioned previously. The following ipptool test will report the current printer attribute values when printing PWG Raster files:

```
{
    VERSION 2.0
    OPERATION Get-Printer-Attributes

    GROUP operation-attributes-tag
    ATTR charset "attributes-charset" "utf-8"
    ATTR naturalLanguage "attributes-natural-language" "en"
    ATTR uri "printer-uri" "ipp://printer.example.com/ipp/print"
    ATTR name "requesting-user-name" "John Doe"
    ATTR mimeMediaType "document-format" "image/pwg-raster"
    ATTR keyword "requested-attributes" "printer-description","job-template","med
}
```

Note: The "requested-attributes" attribute lists attributes (or groups of attributes) that the client is interested in. The 'printer-description' group asks for all status and information attributes while the 'job-template' group asks for all capability attributes. For compatibility reasons, the "media-col-database" attribute needs to be requested explicitly.

The same request using the CUPS API would look like the following:

```
#include <cups/cups.h>

...

http_t *http;
```

```
ipp_t *request, *response;
static const char * const requested_attributes[] =
{
  "printer-description",
  "job-template",
  "media-col-database"
};

http = httpConnect2("printer.example.com", 631, NULL, AF_UNSPEC, HTTP_ENCRYPTION_

request = ippNewRequest(IPP_OP_GET_PRINTER_ATTRIBUTES);
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI, "printer-uri", NULL, "ipp:/
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME, "requesting-user-name", NU
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_MIMETYPE, "document-format", NUL
ippAddStrings(request, IPP_TAG_OPERATION, IPP_TAG_KEYWORD, "requested-attributes"

response = cupsDoRequest(http, request, "/ipp/print");

ipp_attribute_t *attr;
const char *name;
char value[2048];

for (attr = ippFirstAttribute(response); attr; attr = ippNextAttribute(response))
{
  name = ippGetName(attr);

  if (name)
  {
    ippAttributeString(attr, value, sizeof(value));
    printf("%s=%s\n", name, value);
  }
}
```

And this is how you'd query a printer using the nodejs API:

```
var ipp = require("ipp");
var printer = ipp.Printer("http://printer.example.com:631/ipp/print");

var msg = {
  "operation-attributes-tag": {
    "requesting-user-name": "John Doe",
    "document-format": "image/pwg-raster",
    "requested-attributes": ["printer-description", "job-template", "media-col-da
  }
};

printer.execute("Get-Printer-Attributes", msg, function(err, res) {
        console.log(err);
        console.log(res);
});
```

Chapter 3: Print Jobs

What are Print Jobs?

Print jobs in IPP are objects that represent work to be done by a printer. Print jobs provide attributes that describe the status of the job (pending, held for some reason, printing, completed, etc.), general information about the job (the job's owner, name, submission time, etc.) the job ticket (print options), and the job receipt (what print options were used, how many pages were printed, when the job was printed, etc.)

Job Status Attributes

Job objects provide two main status attributes: "job-state" and "job-state-reasons". The "job-state" attribute is a number that describes the general state of the job:

- '3': The job is queued and pending.
- '4': The job has been held, e.g., for "PIN printing".
- '5': The job is being processed (printed, faxed, etc.)
- '6': The job is stopped (out of paper, etc.)
- '7': The job was canceled by the user.
- '8': The job was aborted by the printer.
- '9': The job completed successfully.

The "job-state-reasons" attribute is a list of strings that provide details about the job's state:

- 'none': Everything is super, nothing to report.
- 'document-format-error': The document could not be printed due to a file format error.
- 'document-unprintable-error': The document could not be printed for other reasons (too complex, out of memory, etc.)
- 'job-incoming': The job is being received from the client.
- 'job-password-wait': The printer is waiting for the user to enter the PIN for the job.

  Note: The IANA IPP registry lists all of the registered strings for the "job-state-reasons" attribute.

Page counts are recorded in the following attributes:

- "job-impressions-completed": The number of sides/images that were printed or sent.
- "job-media-sheets-completed": The number of sheets that were printed.

- "job-pages-completed": The number of document pages that were processed.

Job Information Attributes

Job objects provide many informational attributes, including the job's name ("job-name"), number ("job-id"), owner ("job-originating-user-name"), printer ("job-printer-uri"), and page counts ("job-impressions", "job-media-sheets", and "job-pages") which are provided or generated in the job creation request (Create-Job or Print-Job).

Job Ticket Attributes

Job ticket attributes tell the printer how you want the document(s) printed. Clients can query the printer capability attributes to get the supported values. The following is a list of commonly-supported attributes:

- "media": The desired paper size for the print job using a self-describing name (keyword) value. For example, US Letter paper is 'na_letter_8.5x11in' and ISO A4 paper is 'iso_a4_210x297mm'.
- "media-col": The desired paper size and other attributes for the print job using a collection of key/value pairs for size, type, source (tray), and margins.
- "copies": The number of copies to produce. This attribute is typically only supported for higher-level formats like PDF and JPEG, so it is important to specify the "document-format" value when you query the printer capabilities.
- "sides": A keyword that specifies whether to do two sided printing. Values include 'one-sided', 'two-sided-long-edge' (typical 2-sided printing for portrait documents), and 'two-sided-short-edge' (2-sided printing for landscape documents).
- "print-quality": An enumeration specifying the desired print quality. '3' is draft, '4' is normal, and '5' is best quality.
- "print-color-mode": A keyword specifying the color printing mode to use. The value 'color' specifies a full-color print, 'monochrome' specifies a grayscale print, and 'bi-level' specifies a black-and-white (no shades of gray) print.
- "print-scaling": A keyword specifying how to scale the document for printing. This is typically used only when printing images, but can be used for any format. Values include 'auto' (scale to fit or fill as needed), 'auto-fit' (scale to fit as needed), 'fill' (scale to fill the page), 'fit' (scale to fit the document/image on the page), and 'none' (no scaling).
- "printer-resolution": The output resolution to use when printing. The default is usually influenced by the "print-quality" value, but this attribute can be used to force the output resolution to a particular value that is supported by the printer.

- "page-ranges": A list of document pages to be printed, for example '1-8' to print pages 1 through 8. This attribute is typically only supported for higher-level formats like PDF so it is important to specify the "document-format" value when you query the printer capabilities.
- "finishings": A list of enumerations specifying how the output should be finished, for example '3' for no finishing, '4' to staple, '5' to punch, and '4,5' to staple and punch.
- "finishings-col": A list of collections of key/value pairs describing how the output should be finished. Basically an advanced alternative to the "finishings" attribute.

Job Receipt Attributes

Some printers also record a read-only job receipt in attributes named "xxx-actual" for each job template attribute, for example "copies-actual", "media-actual", and so forth.

Documents

Printers report the document formats they support in the "document-format-supported" printer capability attribute. Most IPP printers support standard formats like PDF ('application/pdf'), PWG Raster ('image/pwg-raster'), and JPEG (image/jpeg). AirPrint printers also support a simple raster format called Apple Raster ('image/urf').

Many IPP printers also support legacy formats such as Adobe PostScript ('application/postscript'), and HP Page Control Language (PCL, 'application/vnd.hp-pcl'), along with a variety of vendor-specific languages.

The 'application/octet-stream' document format is used to tell the printer it should automatically detect the format. Detection accuracy varies widely between printers, so you should specify the actual format whenever possible.

Submitting Print Jobs

There are two ways to submit a print job:

- Using the Print-Job operation, and
- Using the Create-Job and Send-Document operations.

The Print-Job Operation

The Print-Job operation allows you to create a print job and send the document data in one request. While all IPP printers support this operation, using it means that you cannot reliably cancel the job while it is being submitted, and for many

document formats this means that the entire job will be printed before you get the response to the request.

The following ipptool test will submit a US Letter print job using the Print-Job operation:

```
{
    VERSION 2.0
    OPERATION Print-Job

    GROUP operation-attributes-tag
    ATTR charset "attributes-charset" "utf-8"
    ATTR naturalLanguage "attributes-natural-language" "en"
    ATTR uri "printer-uri" "ipp://printer.example.com/ipp/print"
    ATTR name "requesting-user-name" "John Doe"
    ATTR mimeMediaType "document-format" "$filetype"

    GROUP job-attributes-tag
    ATTR keyword media "na_letter_8.5x11in"

    FILE $filename
}
```

The same request using the CUPS API would look like the following:

```
#include <cups/cups.h>

...

const char *filename;
const char *filetype;
http_t *http;
ipp_t *request;

http = httpConnect2("printer.example.com", 631, NULL, AF_UNSPEC, HTTP_ENCRYPTION_

request = ippNewRequest(IPP_OP_PRINT_JOB);
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI, "printer-uri", NULL, "ipp:/
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME, "requesting-user-name", NU
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_MIMETYPE, "document-format", NUL

ippAddString(request, IPP_TAG_JOB, IPP_TAG_KEYWORD, "media", NULL, "na_letter_8.5

ippDelete(cupsDoFileRequest(http, request, "/ipp/print", filename));
```

And this is how you'd print a job using the nodejs API:

```
var ipp = require("ipp");
var fs = require("fs");
var printer = ipp.Printer("http://printer.example.com:631/ipp/print");
var filename;
var filetype;

var filedata = "";
```

```
fs.readFile(filename, function(err,data) {
  filedata = data;
}

var msg = {
  "operation-attributes-tag": {
    "requesting-user-name": "John Doe",
    "document-format": filetype
  },
  "job-attributes-tag": {
    "media": "na_letter_8.5x11in"
  },
  data: filedata
};

printer.execute("Print-Job", msg, function(err, res) {
        console.log(err);
        console.log(res);
});
```

The Create-Job and Send-Document Operations

The Create-Job and Send-Document operations split the job submission into two steps. You first send a Create-Job request with your job template attributes, and the printer will return a "job-id" value to identify the new job you've just created. You then send a Send-Document request with your document data to complete the job submission. If you want to stop the job while sending the document data, you can open a separate connection to the printer and send a Cancel-Job request using the "job-id" value you got from the Create-Job request.

The following ipptool test will submit a US Letter print job using the Create-Job and Send-Document operations:

```
{
    VERSION 2.0
    OPERATION Create-Job

    GROUP operation-attributes-tag
    ATTR charset "attributes-charset" "utf-8"
    ATTR naturalLanguage "attributes-natural-language" "en"
    ATTR uri "printer-uri" "ipp://printer.example.com/ipp/print"
    ATTR name "requesting-user-name" "John Doe"

    GROUP job-attributes-tag
    ATTR keyword media "na_letter_8.5x11in"

    EXPECT job-id OF-TYPE integer
}

{
    VERSION 2.0
    OPERATION Send-Document
```

```
    GROUP operation-attributes-tag
    ATTR charset "attributes-charset" "utf-8"
    ATTR naturalLanguage "attributes-natural-language" "en"
    ATTR uri "printer-uri" "ipp://printer.example.com/ipp/print"
    ATTR integer "job-id" $job-id
    ATTR name "requesting-user-name" "John Doe"
    ATTR mimeMediaType "document-format" "$filetype"
    ATTR boolean "last-document" true

    FILE $filename
}
```

The same request using the CUPS API would look like the following:

```
#include <cups/cups.h>

...

const char *filename;
const char *filetype;
http_t *http;
ipp_t *request, *response;

http = httpConnect2("printer.example.com", 631, NULL, AF_UNSPEC, HTTP_ENCRYPTION_

request = ippNewRequest(IPP_OP_CREATE_JOB);
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI, "printer-uri", NULL, "ipp:/
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME, "requesting-user-name", NU

ippAddString(request, IPP_TAG_JOB, IPP_TAG_KEYWORD, "media", NULL, "na_letter_8.5

response = cupsDoFileRequest(http, request, "/ipp/print", filename);

int job_id = ippGetInteger(ippFindAttribute(response, "job-id", IPP_TAG_INTEGER),

ippDelete(response);

request = ippNewRequest(IPP_OP_SEND_DOCUMENT);
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI, "printer-uri", NULL, "ipp:/
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_INTEGER, "job-id", job_id);
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME, "requesting-user-name", NU
ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_MIMETYPE, "document-format", NUL

ippDelete(cupsDoFileRequest(http, request, "/ipp/print", filename));
```

And this is how you'd print a job using the nodejs API:

```
var ipp = require("ipp");
var fs = require("fs");
var printer = ipp.Printer("http://printer.example.com:631/ipp/print");
var filename;
var filetype;

var filedata = "";
```

```
fs.readFile(filename, function(err,data) {
  filedata = data;
}

var create_msg = {
  "operation-attributes-tag": {
    "requesting-user-name": "John Doe"
  },
  "job-attributes-tag": {
    "media": "na_letter_8.5x11in"
  }
};

var job_id = 0;

printer.execute("Create-Job", msg, function(err, res) {
        console.log(err);
        console.log(res);

        job_id = res["job-id"];
});

var send_msg = {
  "operation-attributes-tag": {
    "job-id": job_id,
    "requesting-user-name": "John Doe",
    "document-format": filetype
  },
  data: filedata
};

printer.execute("Send-Document", msg, function(err, res) {
        console.log(err);
        console.log(res);
});
```

Appendix A: Quick Reference

Tools and Libraries

Network debugging tools:

- Wireshark

Libraries and sample code:

- C-based: CUPS and PWG IPP Sample Code
- Java: Java IPP Client Implementation and Core parser for a Java implementation of IPP
- Javascript IPP Client Implementation
- Python IPP Client Implementation

Common Operations

The following table lists the common IPP operations (all defined in RFC 8011) and the commonly-used attributes. Each request always starts with the following three attributes:

- "attributes-charset (charset)": Usually "utf-8".
- "attributes-natural-language (naturalLanguage)": Often just "en", the format is either "ll" or "ll-cc" where "ll" is the two-letter language code and "cc" is the country/region code.
- "printer-uri (uri)": Usually "ipp://address/ipp/print" or "ipps://address/ipp/print".

  Note: The syntax uses the standard IPP data types. Except for Job attributes, all attributes are in the operation group. The "document-format" attribute is optional for the Get-Printer-Attributes operation but highly recommended.

Operation	Required Attributes (syntax)	Optional Attributes (syntax)
Cancel-Job	job-id (integer), requesting-user-name (name)	
Create-Job	requesting-user-name (name), job-name (name)	Job Attributes
Get-Job-Attributes	job-id (integer), requesting-user-name (name)	requested-attributes (1setOf keyword)

Operation	Required Attributes (syntax)	Optional Attributes (syntax)
Get-Jobs	requesting-user-name (name)	my-jobs (boolean), requested-attributes (1setOf keyword), which-jobs (keyword)
Get-Printer-Attributes		document-format (mimeMediaType), requested-attributes (1setOf keyword)
Print-Job	requesting-user-name (name), job-name (name)	Job Attributes
Send-Document	job-id (integer), requesting-user-name (name)	document-format (mimeMediaType), document-name (name)

Common Document Formats

The "document-format" attribute specifies the format of a print file. The following is a list of common formats used for printing:

- "application/pdf": Portable Document Format (PDF) files.
- "application/postscript": Adobe PostScript files (legacy).
- "application/vnd.hp-pcl": HP Page Control Language (PCL) files (legacy).
- "image/jpeg": JPEG images.
- "image/pwg-raster": PWG Raster Format files.
- "image/urf": Apple Raster files (AirPrint).
- "text/plain": Plain (ASCII) text with CR LF line endings (legacy).

Common Job Attributes

The following table lists the common Job attributes that are supported by most IPP printers in the Create-Job and Print-Job operations. You can query the corresponding Printer attributes using the Get-Printer-Attributes operation, just remember to send a "document-format (mimeMediaType)" attribute to get the values for the file format you are using.

> Note: The syntax uses the standard IPP data types. A "1setOf something" is an array of one or more values. The "finishings-ready" and "media-ready" Printer attributes should be used when available, otherwise use the "finishings-supported" and "media-supported" attributes. Similarly, the "finishings-col-ready" and "media-col-ready" Printer attributes should be used when available, otherwise use the "finishings-col-database" and "media-col-database" attributes.

Job Attribute (syntax)	Printer Attribute (Syntax)	Standard
copies (integer)	copies-supported (integer)	RFC 8011
finishings (1setOf enum)	finishings-ready (1setOf enum)	PWG 5100.1
finishings-col (1setOf collection)	finishings-col-ready (1setOf collection)	PWG 5100.1
media (keyword)	media-ready (1setOf keyword)	RFC 8011
media-col (collection)	media-col-ready (1setOf collection)	PWG 5100.3
output-bin (keyword)	output-bin-supported (1setOf keyword)	PWG 5100.2
page-ranges (rangeOfInteger)	page-ranges-supported (boolean)	RFC 8011
print-color-mode (keyword)	print-color-mode-supported (1setOf keyword)	PWG 5100.13
print-quality (enum)	print-quality-supported (1setOf enum)	RFC 8011
print-scaling (keyword)	print-scaling-supported (1setOf keyword)	PWG 5100.13
printer-resolution (resolution)	printer-resolution-supported (1setOf resolution)	RFC 8011
sides (keyword)	sides-supported (1setOf keyword)	RFC 8011

Common Printer Attributes

The following table lists the common Printer attributes that are supported by most IPP printers. You can query them using the Get-Printer-Attributes operation.

> Note: The syntax uses the standard IPP data types. A "1setOf something" is an array of one or more values.

Printer Attribute (Syntax)	Standard
document-format-supported (1setOf mimeMediaType)	RFC 8011
ipp-features-supported (1setOf keyword)	PWG 5100.13
ipp-versions-supported (1setOf keyword)	RFC 8011
job-creation-attributes-supported (1setOf keyword)	PWG 5100.11
operations-supported (1setOf enum)	RFC 8011
printer-alert (1setOf octetString)	PWG 5100.9
printer-alert-description (1setOf text)	PWG 5100.9
printer-geo-location (uri)	PWG 5100.13
printer-info (text)	RFC 8011
printer-input-tray (1setOf octetString)	PWG 5100.13

Printer Attribute (Syntax)	Standard
printer-is-accepting-jobs (boolean)	RFC 8011
printer-location (text)	RFC 8011
printer-make-and-model (text)	RFC 8011
printer-more-info (uri)	RFC 8011
printer-name (name)	RFC 8011
printer-output-tray (1setOf octetString)	PWG 5100.13
printer-state (enum)	RFC 8011
printer-state-reasons (1setOf keyword)	RFC 8011
printer-strings-languages-supported (1setOf naturalLanguage)	PWG 5100.13
printer-strings-uri (uri)	PWG 5100.13
printer-supply (1setOf octetString)	PWG 5100.13
printer-supply-description (1setOf text)	PWG 5100.13
printer-supply-info-uri (uri)	PWG 5100.13
printer-uri-supported (1setOf uri)	RFC 8011
pwg-raster-document-resolution-supported (1setOf resolution)	PWG 5102.4
pwg-raster-document-sheet-back (keyword)	PWG 5102.4
pwg-raster-document-type-supported (1setOf keyword)	PWG 5102.4
uri-authentication-supported (1setOf keyword)	RFC 8011
uri-security-supported (1setOf keyword)	RFC 8011

Standards

The IANA IPP Registry provides a list of all IPP attributes, values, operations, and status codes with links to the corresponding standards.

These are the core Internet Printing Protocol standards:

- IPP Everywhere (PWG 5100.14)
- IPP 2.0, 2.1, and 2.2 (PWG 5100.12)
- IPP 1.1 - Encoding and Transport (RFC 8010) - Model and Semantics (RFC 8011)
- IPP Implementor's Guide (PWG 5100.19)

These are the standards for media naming and the common file formats:

- Media Names (PWG 5101.1)
- Portable Document Format (ISO 32000-1)
- PWG Raster Format (PWG 5102.4)

These are the Internet Printing Protocol standards that define how to support specific printer features:

- Alerts (PWG 5100.9)
- Cloud Printing (PWG 5100.18)
- Extended Options - Set 2 (PWG 5100.11) - Set 3 (PWG 5100.13)
- Fax Output (PWG 5100.15)
- Stapling, Folding, Punching, and Other Finishings (PWG 5100.1)
- Notifications - Model (RFC 3995) - Get-Notifications Operation (RFC 3996)
- Output Bins (PWG 5100.2)
- Page Overrides (PWG 5100.6)
- Paid Printing (PWG 5100.16)