INTERNET-DRAFT

HTTP 1.1 Transport Mapping for the Internet Printing Protocol
Missing parts

When Randy wrote the protocol document, he changed to syntax from the binary version that we had previously agreed to. I expected the old version to be in the document as an alternate solution, but it is not. I am including it here.

The following protocol primitives are defined:

DIGIT = "0".."9"

ALPHA = "A..Z"

BYTE = %d0..%d255

OCTET-STRING = *BYTE

The following ABNF specification describes the encoding of an IPP message (PDU):

(We may want to state that string values are case sensitive, unlike what ABNF says).

operation-encoding = operation-version operation operation-data

operation-version = operation-major-version operation-minor-version

operation-major-version = %x00..%x7F ; major version in binary, initially 1

operation-minor-version = %x00..%x7F ; minor version in binary, initially 0

operation = operation-length operator *operand

operation-length = %x00.00.00.00 .. %x7f.ff.ff.ff ; number of octets for the operation in binary

operator = %d0.9 “operation” value-length value

operand = name-length name value-length value

name-length = %x00.00..%x7f.ff ; number of octets of the name in binary

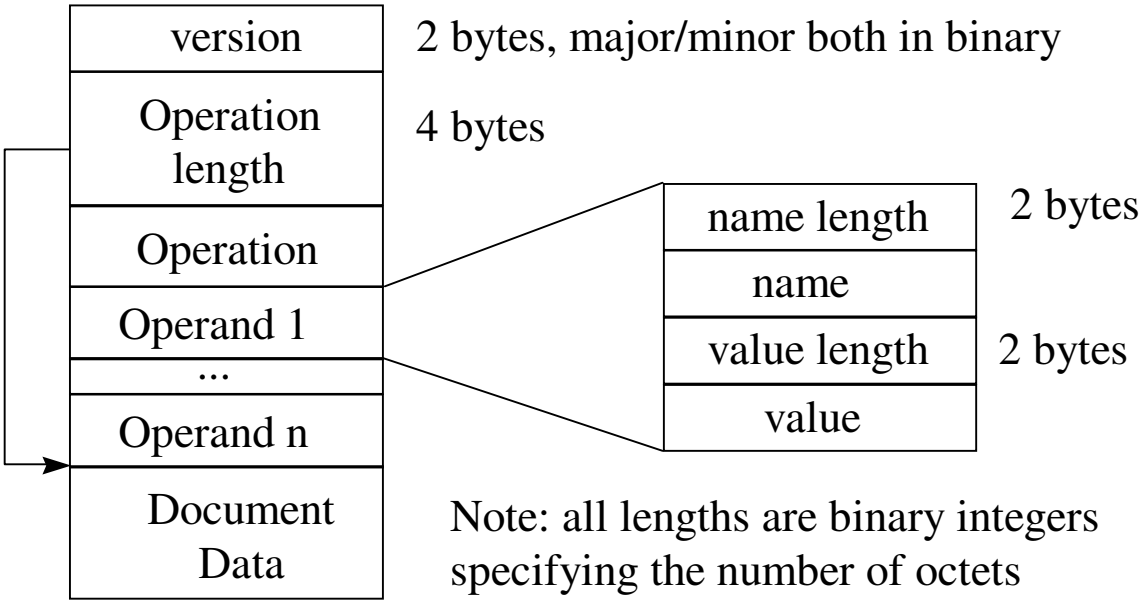
name = = ALPHA *(ALPHA / DIGIT / "_" / “-“)

value-length = %x00.00..%x7f.ff; number of octets of the name in binary

value = OCTET-STRING

operation-data = OCTET-STRING

The following is a picture of this encoding:



The transport encoding in Randy's Appendix A references syntax items that aren't in his document. Here is the new version of the transport encoding for raw TCP/IP that reference items in the operation-encoding.

The idea behind this is that the transport encoding does the work of HTTP. It encodes HTTP header information in the transport-operands. It encodes the application/ipp entity body in the transport-operation by breaking it into chunks.

transport-encoding = transport-version XID transport-operands transport-operation

transport-version = transport-major-version transport-minor-version

transport-major-version = %x00..%x7F; major version in binary, initially 1

transport-minor-version = %x00..%x7F; minor version in binary, initially 0

XID = %x00.00.00.00 .. %x7f.ff.ff.ff ; transaction id in binary

transport-operands = operand-length *operand

operand-length = %x00.00.00.00 .. %x7f.ff.ff.ff ; number of octets for the operands in binary

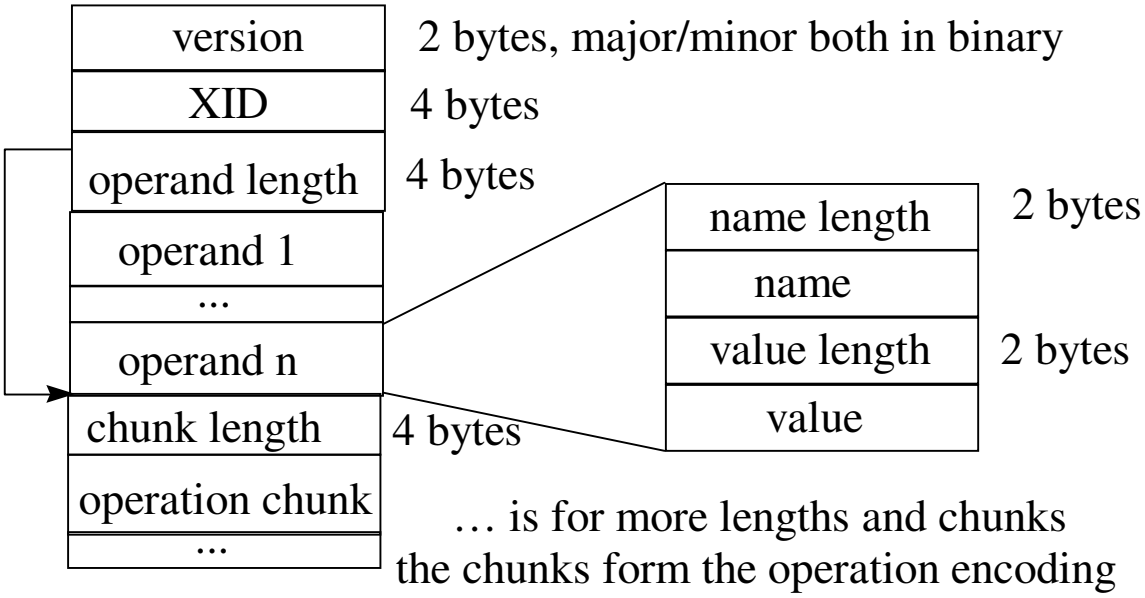
transport-operation = 1*transport-chunk ; the chunked operation-encoding data

transport-chunk = transport-data-length transport-data

transport-data-length = %x00.00.00.00 .. %x7f.ff.ff.ff ; number of octets of the data in binary

transport-data = OCTET-STRING

The following is a picture of the transport encoding:


version
2 bytes, major/minor both in binary
XID
4 bytes
operand length
4 bytes
operand 1
...
operand n
chunk length
4 bytes
operation chunk
...
name length
2 bytes
name
value length
2 bytes
value
… is for more lengths and chunks
the chunks form the operation encoding