**Request for Comments:** CP372 A1 Protocol

**Authors:** Troy Nechanicky, nech5860, 150405860 | Frank Khalil, khal6600, 160226600

**Date:** January 26, 2018

**Description:**

This is a protocol for a multi-client single-server application that supports storing book records.

Record components: ISBN-13, title, author, publisher, year

Clients:

* Request a connection with the server
* SUBMIT, UPDATE, GET, and REMOVE records
  + ISBN is required for SUBMIT and UPDATE requests. If an ISBN is included in any request, the client ensures that it is valid

Server:

* Accepts requested connections and supports multiple simultaneous connections
* Maintains a list of records submitted by clients. It is not persistent. All clients interact with the same list
* Processes messages received from a client:
* SUBMIT: add the given record to the list (if the ISBN is unique)
* UPDATE: update a relevant record in the bibliography list
* GET: send to the client a list of records matching the request. Client may request list in BibTeX format
* REMOVE: remove the specified record from the list

**Client Protocol Format:**

All requests are terminated with an empty line. All elements a request are terminated with a newline character (‘\n’), so ‘\’ (like “\\n”) must be used as an escape character in order to send ‘\n’ as content.

Request connection:

CONNECT

*Note:* The client expects to see a request like CONNECT <IP address> <port number>, then it attempts to send CONNECT to <port number> at <IP address>.

Bibliographic information is expected in the following format (order does not matter):

ISBN <ISBN>

TITLE <title>

AUTHOR <author>

PUBLISHER <publisher>

YEAR <year>

SUBMIT/UPDATE:

<request type>

<bibliographic information>

*Note:*

* A valid ISBN is required for all SUBMIT/UPDATE requests.
* The ISBN in a SUBMIT request must be unique.
* An ISBN cannot be updated through an UPDATE request.

GET:

GET

ALL

BIBTEX

<bibliographic information>

*ALL:* request effects all records

*BIBTEX:* asks server to return records in the BibTeX format

*Note:* If ALL is not used, then at least one of ISBN, TITLE, AUTHOR, PUBLISHER, and YEAR must be specified.

REMOVE:

REMOVE

ALL

<bibliographic information>

*Note:* If ALL is not used, then at least one of ISBN, TITLE, AUTHOR, PUBLISHER, and YEAR must be specified.

The server requests confirmation before records are removed. A response of “Y” confirms, while a response of “N” cancels the remove.

*Note:* A CLOSE request causes the client to close its connection to the server.

**Server Protocol Format:**

All requests are terminated with an empty line.

Acknowledge/Accept request:

SUCCESS: Connection accepted

SUBMIT:

1 of:

SUCCESS: Record added  
ERROR: Duplicate ISBN

UPDATE:

SUCCESS: Record(s) updated

GET – default:

SUCCESS: <x> records retrieved

<bibliographic information>

GET – BIBTEX:

SUCCESS: <x> records retrieved

@book{first-4-char-of-title+year,

isbn={<isbn>},

title = {<title>},

author = {<author>},

year = {<year>},

publisher = {<publisher>}

}

*Note:* Any unavailable characters in the BibTeX key will be replaced with ‘x’.

*Note:* Additional records appear immediately under the first record in the requested format.

REMOVE:

Please confirm the removal of <x> records (Y/N)

Then 1 of:

SUCCESS: Records removed

SUCCESS: Remove cancelled

Errors messages for all requests:

ERROR: Invalid request <request>  
ERROR: Invalid field <field>  
ERROR: Invalid ISBN  
ERROR: Request timed-out after 2 seconds. No empty line signaling end of request received

Error messages for UPDATE, GET, REMOVE:

ERROR: Matching record not found

**Synchronization Policies:**

Only 1 thread at a time is allowed to execute in the sections of code that modify records (i.e. creating, updating, or removing a record). If such an operation is occurring, threads trying to retrieve information for a GET request are blocked.

**Error Handling:**

Errors that the client generates:

* ERROR: Unable to connect to IP address <IP address> at port <port>
* ERROR: Invalid ISBN
* ERROR: Connection to server lost
  + This occurs if the client is unable to send the request
* ERROR: Request timed-out after 2 seconds. The server is likely offline
  + This occurs if the request is sent, but no complete response is received

Uncaught client errors are printed to the console/GUI. Uncaught server errors are printed to the console.

If the connection to the client is lost when the server is trying to send a response, the server will gracefully terminate the operation.

**Server-side Data Structures:**

Bibliography:

* List of book records. Empty when server is started
* Ensures ISBNs of books are unique

Book:

* An array containing the attributes of a book record
* Attributes: ISBN, title, author, publisher, year. ISBN must have a value.