

PRL

TECHNICAL NOTE

TN-85-50

COMPUTERISED BOOK PROCUREMENT
SYSTEM - IMPLEMENTATION

By

C.V.Rajan, R.R. Bharucha and
D.R.Kulkarni

1985

Physical Research Laboratory
Ahmedabad-9.

DOCUMENT CONTROL AND DATA SHEET

-
1. REPORT NO. : PRL : TN-85-50
-
2. TITLE AND SUBTITLE : "Computerised Book Procurement System - Implementation
3. REPORT DATE: Dec., 1985
4. TYPE OF REPORT : Technical Note
5. PAGES : 49
6. PRICE : Unpriced
7. NO.OF REFERENCES : 7
-
8. AUTHORS : C.V.Rajan, R.R. Bharucha & D.R. Kulkarni
-
9. a) PURPOSE : Library Automation
- b) USEFUL FOR : To accelerate process of Book Procurement and to produce useful managerial reports.
-
10. ORIGINATING UNITS/DIVISIONS AND ADDRESS :
Library
Physical Research Laboratory
Navrangpura
Ahmedabad 380 009
-
11. SPONSORING AGENCY : Nil
-
12. ABSTRACT : This document is a sequel to our previous Technical Note which gives details of systems Analysis for Computerised Book Procurement System. This report contains details regarding the implementation of the same system.
- The system not only monitors the status of the request but also produces no.of useful documents and reports, thereby aiding substantially in the procurement of books.
-
13. KEYWORDS : Library Automation, Book Acquisitions
-
14. DISTRIBUTION STATEMENT : Distribution Unlimited
-
15. SECURITY CLASSIFICATION: Unclassified
-

COMPUTERISED BOOK PROCUREMENT SYSTEM - IMPLEMENTATION

C.V.Rajan, R.R. Bharucha and D.R. Kulkarni

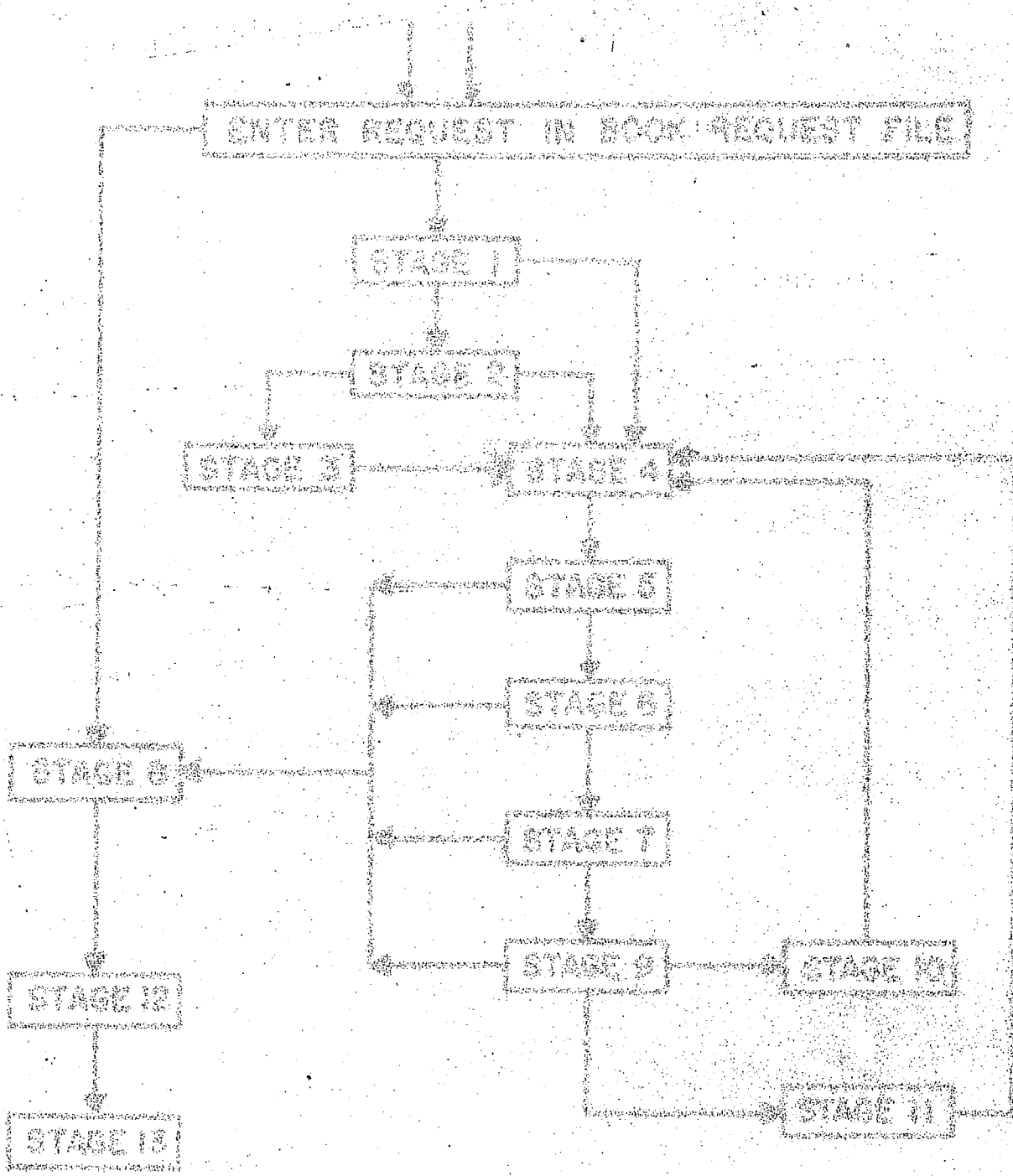
1. What is BPS ?

BPS (Book Procurement System) is a well organized and efficient way of procuring books by the help of which the procurement and processing time can be substantially minimized.

The life span of a book request has been split up into 13 distinct stages (see figure 1). At any instant of time we can find various requests received by the library at different stages of processing. If the rate of receiving requests is reasonably high, the handling of the requests will become unwieldy. However by using computers in handling the requests we can save processing time and effort.

The system has been developed on 7 Disk-based files. The BPS has been implemented using the detailed Systems Analysis carried out in PRL Technical Note⁽¹⁾. BPS is another step forward in the automation of PRL Library.

BPS has been implemented as an online user-friendly system catering to the needs of book requestee and the library staff. Programs have been written such that the users get quick response for their queries on their terminal. Provisions have been made to retrieve information through the unique request number, name of Requestee or Surname of (first or subsequent) author(s) and from words in the title. The string



VARIOUS STAGES DEFINED IN THE BPS SYSTEM

matching algorithm has also been used to retrieve information using any known string present in the request-record.

2. How does it work ?

As seen in figure 1, the request goes through a number of stages. We have briefly described here how this processing proceeds.

2.1 Stage-1

The life of a book request starts when the library staff member puts the request into the system. For putting a request into the system, the minimal information required are the Author's name and title. Besides this, user may also feed the information such as Requestee's name, his section, publisher code of book vendor code (if vendor is known), edition number, binding information (Hard-bound or Soft-bound), number of copies required, price of the book etc. At the time of entering the request, the system performs full field-checking and validation of the input.

Before a new request for procuring a book is entered into the BPS, the system automatically checks to avoid duplication using surname of the first author. This checking is done against both the pending orders and the books which are procured but have not been cataloged. As both this information is contained one file, this checking is done quickly and easily. Again further checking to avoid duplication is done from the READFAST database, which comprises of the main library collection.

The requestees can retrieve information through the request no., name of requestee and by using surname of any of the authors of the book. If author reader does not know any of the above mentioned details, but remembers a substring (i.e. any word) of the title, he can still use it to retrieve the complete details of the request.

After receipt of the books, the BPS data is updated with call no. and keywords and then transferred to the database comprising of the total library collection. This is extremely useful as the inputting is done only once.

2.2 Stage-2

When sufficient requests in stage-1 have piled up, the system will generate a query letter. Each query is identified by a unique query number. The number of the last query letter sent will be stored in the header record of request-file. From this we can find the number which is to be given to the present query letter. After sending the query letter, the header-record will be suitably updated. Provisions have been made (using the conditional compilation facility of DEC-10 FORTRAN) to generate query letter on terminal or printer. After generating query letter on the printer, the stage of each request included in the query letter will be changed from 1 to 2 and the query number in the header-record will be updated. However, if the query letter is generated on terminal, the header-record won't be updated and the stage won't be changed to 2. This helps in having a final look at the query letter before printing the same, so that some modifications (like spelling errors in Author name and title) can be made if required. As generating query letter on terminal doesn't make any updates in the request-file, we can generate it on terminal as many times as we wish. This will also be helpful for demonstration purposes, provided there are some requests in stage-1.

2.3 Stage-3 & 4

After sending query letter, we will receive various kinds of replies from vendors. Some vendors may inform us that they have some books in ready stock while some others may inform us that some books are out of print or out of stock, or that the book is not yet published. Sometimes the information included in our query letter may be in-sufficient and the vendor may write to us for more particulars about the book.

All this information has to be fed into the system so as to make the update current.

For updating the Request file using the vendor replies, for the query letter, we should know the relative position of the vendor in the vendor file. Sometimes more than one vendor may inform us that they have the same books in ready stock. In that case we will select the vendors on the FCFS (First Come First Served) rule. Provisions have been made to store upto first 3 vendor codes in the request file. In the request file we keep a variable OSW (Order Status Word) which indicates the status of the order. If atleast one vendor informs us that the book is in ready stock, we will set this variable to 'CONF' to indicate confirmation of the order. If the book is out of print, we store 'OP' and if the book is out of stock we store 'OS' in this variable.

If the book is a new publication (yet to be published) we store 'NPUB' in this variable. If the vendor requires more particulars, we store 'MPR' in OSW. However, note that all these possibilities are not mutually exclusive. If the OSW has been set to 'CONF' by the reply of one vendor, system will not permit one to alter it to 'MPR' or 'OS' by using the reply of another vendor. However, the OSW containing responses other than 'CONF' can be changed subsequently. The relative position will be stored in the Request file, only for 'CONF' & 'MPR' replies. There is no advantage in storing the vendor codes if the vendor reply is 'OP' or 'OS'. If atleast one vendor is known for a book, the stage of the request will be made 4. If we do not receive a reply for some of the books in the query within the stipulated period of two weeks, the stage will be changed from 2 to 3.

2.4 Stage-5

If there is atleast one request in stage 4, the system is ready to prepare Indian order letters to be sent to the first of the three vendors stored in the Request file. Each order is identified by the unique order number given to it and the date on which it was sent. The total number of orders sent will be stored in the header-record of request file.

Provisions have been made to generate the order letters on terminal as well as on printer. This helps in rectifying errors and also for demonstrating the working of the system. When order letter is generated on terminal, no updates will be made in the request file. However, when order letters are generated on printer, all stages of requests included in the order letter will be changed from 4 to 5 and the order number will be incremented by 1 and header record will be modified accordingly. We will also store the order number and date on which order was sent in the request record. If by chance the vendor to whom our first order letter had gone is unable to supply the book, we will check the request file if there is a second vendor for that book. If there is one, we will send a new order letter to this vendor and so on.

Foreign orders will be sent if the Indian vendors are unable to supply the book in time. All requests in stage 3 and all requests in stage 5 for which the book is not received within the stipulated time and all books in stage 9 which are to be re-ordered are eligible for Foreign orders.

The foreign orders can be sent either directly to publisher or to our foreign vendors or to special agencies (like Foreign Universities, directly to Authors etc.).

We set a variable ORDIND to 'I' if the Indian order is sent and to 'F' if the Foreign order is sent to and to 'S' if order is sent to Special Agents. We can generate foreign order letters either on terminal or on printer. It may be noted that the decision to generate a foreign order letter (to whom it is to be addressed) is to be taken by the library staff.

2.5 Stages 6,7,8 & 9

After sending the foreign orders, we receive various kinds of replies as in the case of Indian orders. The system will update the request file using these replies. When we send the foreign orders, we store the relative position of the publisher (in the Publisher File) in the request file. We should also know after how many days the first and second reminders have to be sent. These will also be stored in the request record.

If we do not receive a reply for our order letter, within the stipulated time, we will generate the first reminder letter and change the stage to 6. If a reply comes in between, we will immediately update the request file to preclude the generation of further reminders. If no reply is received, for the foreign order and first reminder, we will generate the second reminder letter. After sending second reminder, the stage will be changed from 6 to 7. If the book is received

the stage will be changed to 8 and we will feed

all other items like price, publication year etc. and we will check for errors in Author name, publisher code, edition etc.

A request in stage 7 enters stage 9 when the book has not been received by the library even after sending the second reminder. When the request reaches stage 9, the system will be helpless to process the request further, without the help of library staff. The library staff can either cancel the request or re-order the book by selecting some other vendor. The new vendors relative position and other details have to be fed into the system and the stage has to be reset to 4 at this time.

When the library staff decides to cancel the request, the system will set the variable 'CODE' in the request file to '*' so that the system can easily detect all cancelled requests. The cancelled requests have to be deleted from the system periodically. After deleting all cancelled requests, we should update (or re-create) the associated files.

2.6 Stages 10, 11 & 12

Stage-10 : It is a stage when the order placed for a book has been cancelled.

Stage-11 : It is a stage when the system has been informed by the supplier that the requested book is not available on the regular market. We will set the order status word

to 'OP' or 'OS' accordingly. At this stage we ask the requestee about the action to be taken for further processing of the request.

Stage-12

The request in stage 8 enters this stage when all the details of the book such as accession number, classification numbers, keyword, actual price, period etc. have been supplied to the system and the book is ready to be issued. This is in brief the discussion of how BPS works. The chapter 2 describes various files, their structure and contents used in BPS.

Stage-13

After all details of the book had been fed in, we will make an intermediate file from the Request file, which will later be loaded into the READFAST database. The fields in this intermediate file will be compatible with the fields in the database files of READFAST. While forming the intermediate file, we will keep additional fields for keywords, classification numbers etc. This helps the library in carrying out various analyses during each financial year. The library can understand the amount spent subjectwise in each financial year, amount spent on each requestee, amount spent on different sections in each financial year etc. The intermediate files will be kept till the end of each year, and various reports necessary for the library will be produced from it before it is deleted.

1. File Structure in BPS

BPS is based on 7 files. As the Request File and Publisher File will be accessed quite often they have been created as Direct Access Files. The other files are all sequential and can be stored on Disk or Tape. Figure 2 show various files used in BPS.

The request file contains details of the request. It is the most important file in BPS. The Publisher File contains details about the publishers. The Vendor File contains the details of Indian Vendors. The Department/Section file provides identification of the Requestees. The associated files are formed automatically and are used for retrieval purposes.

Associated File-1 is used for retrieving the request record using first K letters of requestee. (K can be conveniently chosen as 3). This file stores the first 3 letters of Requestee's name and the corresponding relative position in the Direct Access Request File.

The Associated file-2 contains the first three letters of author's name and the corresponding relative positions. This will also be used for retrieval purposes. The process of retrieving is explained in the information retrieval part. To avoid frequent internal sorting, we will keep the Associated file-1 sorted externally according to alphabetical order of Requestee's name and Associated file-2 sorted externally according to Author's name. The Associated File-3 can be kept sorted externally according to ascending request number.

2. Contents of Files

a) The Request File:

This is the most important file in BPS. It is a Direct Access (DA) file. This file contains the crucial information about each request. The various fields are given below:

- i) Code : The Request can be Personal, Official, Student Book Grant etc.
(1 Alpha-bet)

Sometimes the Library receives gratis copies from various sources.

CODE: P - Personal
O - Official
S - Student Book Grant
G - Gratis copy
B - British Council Grant
U - USO

- ii) Request # : Each request is assigned a unique request number for ease of retrieval.
(8 Numeric)

The request no. can be formed in various ways. However the simplest way is to give a serial no. between 1 and 99 for each request received on any particular day and to concatenate this number with the date, month and year (2 digit) to form the request no. e.g. suppose we receive 30 requests on 12th Aug.85. We can form the last request no. as

30 12 08 85

DP MM YY

- iii) Title (125 Alpha-numeric) : The Book Title is assigned a field width of 125 characters.
- iv) Publisher Code (2 numerals + 6 Alphabets) : Each publisher is assigned a publisher code comprising of the country code and publisher-name code.
- v) Author (40 Alpha-numerals) : 40 characters are set aside for the Author name. If there is more than one author, each is separated by a delimiter character blank.
- vi) Edition (6 Alpha-numeric) : The books can come in different editions and could be either hard-bound or soft-bound and are published in different years. The edition field is formed by concatenating the edition no., H or S (for Hardbound/Softbound) and year of publication e.g. for a book in its 4th edition and softbound published in 1985 we give edition as 4S1985.
- vii) Copies (1 Alpha-numeric) : If we get more than one request for the same book, simultaneously, we put only one request into the system and modify the COPIES field appropriately. However for requests for the same book received at different times, we put different requests into the system.

- viii) Stage (Integer) : As described in the Systems Analysis the life of each request is broken down into 13 stages. This field indicates in which stage the request is at any particular time.
- ix) Price (Real *8) : This field stores the price of the book in Rupees.
- x) Update Day
Update month
Update Year
(Integer *4) : For different books, the order letters, reminder letters etc. will be sent on different days. To have an idea about this, we store the last update day, month and year of Request file in these fields. This date will be used for changing the stage of the request from 2 to 3 and from 7 to 9.
- xi) Vendor Code 1
Vendor Code 2
Vendor Code 3
(Integer *4) : After sending query, we may receive positive replies for the same book from different vendors. We store the vendor codes of these vendors in a FCFS manner in these fields. Order letter will be sent only to the first vendor. If he is unable to procure the book in the specified time, we will order from the second vendor and so on.

- xii) Requestee's Name (20 alphabets) : We store the Requestee's name and section in these fields
- xiii) Section of Requestee (5 alphabets)
- xiv) OSW(Order Status Word)(4 alphabets) : The various decisions to be taken on a request will be indicated by the field OSW. When the request is entered into the system, OSW is initialized to some value. Afterwards it may assume any of the following.
- CONF : Supply of the book is CONFirmed
- OP : Book is Out of Print
- OS : Book is Out of Stock
- DESP : Book is already DESPatched
- REOR : Book is to be REORdered
- ADV : Advance payment is required for further processing.
- xv) Query Number : These fields contain the query
Order Number
Reminder Number
(Integer *4) number, order number etc.
- xvi) Accession Number : When the book is received in the
(Integer *4) library, we will give the acces-
Call Number sion no. and call no. to it.
(Character *10) Before the book is received in the library, we would not keep these fields idle. The field for CALL

Number will be used for storing the Reference Number of Vendors reply (if the vendor has any reference no. at all). The field for Accession no. can be used, if required, for some other purpose till the book is received.

- xvii) Advance Payment (Real *8) : Some foreign publishers require advance payment, before sending the book. We will store the advance amount paid in rupees in this field.
- xviii) ORDIND (Order Indicator) (character *1) : The orders can go either to Indian vendors or foreign vendors or to special agencies like foreign universities, institutes, or directly to the author of the book etc.
- ORDIND = (I if order goes to Indian vendor
(F if order goes to Foreign Vendor/Publisher
(S if order goes to Special Agencies

If the order is placed to Indian Vendor, we will store the relative position of the vendor's address in the vendor file in the field vendor code 1. We will store the second and third preference of the vendor in vendor code 2 and vendor code 3.

However, if the order is placed to a foreign vendor or publisher, we store the relative position of the vendor or publisher in vendor-code-1 and no. of days after which first reminder is due in the field vendor-code-2 and no.of days after which second reminder is due in the field vendor code 3.

Thus depending upon the field ORDIND the fields vendor-code-2 and vendor code 3 contain different entities.

If we send the order to some special agents, we will store the address of the special vendor in a separate file and the relative position of the address in this file will be stored in vendor-code 1 of the Request record. This will help us in keeping the Indian Vendor file and Foreign Publisher file intact and free from frequent updates.

Each record in the request file contains 303 characters, according to the current design. This file can contain a maximum of 1000 records as per current specifications in the program.

b) Department/Section File:

The file is kept for identification of requestee and for sending intimation letters to requestees.

- i) Section-code : Each Department or Section is (5 characters) identified by a code. In the request file, we will store the section code of each requestee along with the name.

ii) Dept. Name : We will store the Department
(30 characters) name or other identification
in this field.

The file contains at present
12 records.

c) Vendor File : The status code stores the
Status Code status of the vendor.
(1 character)

Vendor code : The serial number of vendor in
(Integer *4) Vendor File.

Vendor Name : This field stores the name of
(40 Characters) vendor.

Vendor Address : We will store the address of the
(80 characters) vendor in this field.

Total orders,
Orders cancelled,
Total Query,
Total Reminders,
Total Books supplied : To have an idea about the vendor
(Integer *4) performance, we will store total
orders sent to a vendor, orders
cancelled, total query letters
sent, total no. of reminders
sent etc. for each vendor.

Mean response time : As the response time for our
Standard deviation queries and orders varies from
of response time vendor to vendor, we will store
(Real *8) the cumulative mean and standard
deviation of response times for

each vendor. Whenever a new response comes from the vendor, we will update these fields to have the cumulative mean and standard deviation.

d) Publisher File

The publisher file contains information about all foreign publishers and foreign book suppliers. The various fields in the Publisher file are given below :

- i) Indian Agent Code (Integer) : Each publisher abroad may have a representative in India. This field stores the relative position of the Indian vendor in the vendor file who is the most appropriate representative in India. If a publisher has more than one representative in India, we will store only the most appropriate one. This field will help us in choosing the appropriate books to be included in the Query letter to any Indian vendor. For example, Narosa Book Distributors deals exclusively with books published by Springer-Verlag, and few other foreign

publishers. While preparing Query letter to them, we need include only those books whose publishers are represented by them.

- ii) Publisher Code : Each publisher will be assigned a
 (2 numerals + publisher code by the library
 6 Alphabets) staff, which comprises of the
 country code and the publisher
 name code. In the Request file,
 against each book we will store
 only the publisher code.
- iii) Publisher Name : This field contains the name of
 (40 alphanumerals) the publisher. If more than one
 publisher in the same country,
 has the same name, we will give
 different publisher name code to
 distinguish between them.
- iv) Address : This field has been split up into
 (Street- 30 character-
 ers
 City - 40 " STREET, CITY, COUNTRY and ZIP.
 Country-30 " It is most important to keep
 Zip - 10 " these fields accurate because
 the foreign orders will be sent
 using this address.

- v) Discount (Integer *4) : If the publisher offers any discount, we will store the discount in this field.
- vi) Privileges (Character *8) : Some publishers may require some advance payment while some others may require full pre-payment before despatching the book. We will store this information in this field.
- vii) Total orders placed (Integer) : These fields will be used for statistical analysis of each publisher.
Total replies received (Integer)
Total amount dealt with. (Real *8)

CHAPTER - 3

CREATION AND UPDATES OF FILES

In chapter 2 we have seen the fields of various files in BPS. In this chapter we will see how they are created and updated using BPS modules. In this discussion we will ignore the Associated File-1, 2 & 3 as they are generated and updated automatically.

1. Creation of Request File

a) To create the request file for the first time, it is necessary to create the header-record for it. To create the header-record type the command `RUN RECRAT`.

This program will ask for the details such as total number of requests in the file, total number of different documents sent, total books acquired by PRL Library etc.

b) After creating the header-record, the request file is created by adding records one by one using the program `CRQ` described later.

Updating the request file

a) Addition

Type the command

`.RUN CRQ`

Details of the record to be added are asked for interactively.

b) Deletion

Type the command

```
.RUN DELREQ
```

The request to be deleted is accessed through author-name or any substring in the author-name field. Then the same can be deleted by putting a flag '*' in the request code.

c) Modification

Type the command

```
.RUN MODIFY
```

The request to be modified is accessed through author name or any substring in the author-name or any substring in the title field. After accessing the desired record, the program displays all the field names and the corresponding field numbers. The fields to be modified are to be indicated by the field numbers separated by comma. The current contents of the field will be displayed and the new value will replace the previous value in these fields.

In BPS, frequent updates are made in the request file whereas only occasional updates are made in other files. As the associated files can be recreated from the request file, there are no updates made directly in these files. If there is duplication of request numbers in the request file, retrieval using request numbers may be erroneous.

In such cases we modify the request file after, eliminating duplicate request numbers and re-create the associated files.

i) Request File:

There are 2 types of updates and modifications made in the request file.

- a) Routine updates
- b) Occasional updates

a) Routine updates

When reply comes from vendors or publishers for our query, orders or reminders, we will have to update the request file to make the information contents of these files current. Likewise when vendor requires more particulars we will have to get the additional information from book requestee's and this will have to be fed into the system. Similarly, sometimes the requestee may wish to change the binding information of the book to get cheaper editions or they may take a decision to cancel the request while the request is in the process of procurement. These are the routine updates in request file. We will use one program (UPD.FOR) for processing replies from vendors and publishers and another program (MODIFY.FOR) for modifying any desired field of the request file.

b) Occasional updates :

Sometimes when the process of generating order letters or reminder letters is in progress, we may make an erroneous input and this may result in wrong updates of the request file. For example, at the time of generating foreign order letters, the library staff has to decide whether the order has to be sent directly to publisher, or to our foreign vendors or to special agents etc. If we make a wrong input at this stage, the system will put the information wrongly in the new request file. Likewise if there is a power failure when the process of generating orders etc. are in progress, the request file may be partially updated and we may have to modify some records to proceed with our task. Similarly due to problems in taking hard copies, we may have to generate one more hard copy for a previously generated document. In such cases, we will have to modify a few fields of the request file. These are the occasional updates.

In occasional updates, depending on the prevailing situation, we will have to decide which are the fields to be modified and what are the new values to be given to these fields. For example, before generating foreign orders, the stage of a request may be 1,2,3,4 or 9 and after generation it will be made 5. Thus if we want to generate one more order letter for a previously generated order, we will have to take the records of those requests into their previous states before generating order. Then only the

requests will satisfy the conditions for the generation of order. These can be accomplished by finding the relative positions of the records to be modified in the request file and then making use of the program MODIFY.FOR.

The number of the last Query letter generated, last order, reminder etc. generated will be stored in the header-record of the request file. Similarly we will store the total number of books in PRL library.

2. Creation and updates of Vendor-file and Publisher-File

a) To create the vendor file, type the command

```
.RUN CREVEN
```

The program will ask for all details about the vendor one by one and create the vendor file.

b) To update the vendor file, type the command

```
.RUN UPDVEN
```

This program will at first locate the vendor record using the relative position inputted and do necessary updates one by one.

c) Creation and updates of the Publisher file are done by a single program. Type the command .RUN MODPUB

The program will ask for the publisher code. Using this publisher code, the system will do a binary search on the sorted publisher codes and corresponding relative position arrays and retrieve the publisher record to be modified. If the publisher code does not exist in the

publisher file, the system will create a new publisher record at the end of the existing publisher file. We will have to update the vendor file when the vendor changes his address or we have to add a new vendor. Each vendor is identified by the relative position in the vendor file. As noted previously, the fields vendor-code-1, vendor-code 2 etc. in the request file contains the relative positions of different vendors on priority basis. Thus the updates in the vendor file should not disturb the relative positions of existing vendor records. Addition of new vendors are made at the end of the vendor file.

As the publisher file is organized as a direct access file and as each publisher is uniquely identified by the publisher code, updates and modifications in the publisher file can be made at any desired position. While assigning publisher codes to new publishers, care should be taken to see that the given code does not conflict with any of the previous publisher-codes and that the codes given to foreign vendors.

CHAPTER - 4INFORMATION RETRIEVAL AND GENERATION OF DOCUMENTSInformation Retrieval

The most important aspect of BPS is the Information Retrieval and Generation of various documents. For retrieving information from BPS some pre-defined queries have been developed. The queries can be broken down as follows :

- i) Queries to be answered through IQL package
- ii) Built in Queries to be answered directly by BPS

IQL Queries

An IQL Dictionary has been created for the request file to answer the following queries :

- i) List all requests using Requestee's name
- ii) List all requests using Author's name
- iii) List all requests in a given stage
- iv) List all requests in all stages

For each query a MIC (Macro Interpreted Command) file has been created, so that a user having no knowledge of IQL can also retrieve information for his query. The details of these files and their usage are given in Appendix - 1. It may be noted that IQL has been used only for retrieval from request-file and hence only READ-PASSWORD is operational.

Built-in Queries

Programs have been written to retrieve information contained in the request using the Unique Request-number, Requestee's name, the Surname of the first Author, any other known substring contained in the request record or any combination of code, stage, OSW and publisher field.

i) Using Request-number

Each request is uniquely identified by the Request number assigned to it by the library staff. By using the request number we can retrieve the request record easily.

ii) Using Requestee's name

As pointed out earlier, the Associated-file-1 contains the first 3 letters of Requestee's name and the corresponding relative positions. We will do a Binary search on the sorted Associated-file-1 and stack the relative positions of all records having first 3 letters common. Afterwords we will read the request record corresponding to these stacked relative positions and check if the rest of the requestee's name is present in the "Requestee's name" field of these records. If it turns out to be a success, we will display the record.

Thus by a correct input of Requestee's name, the recall-ratio in retrieving the request record is maximum. However if the rest of the requestee's name is not present in any of the request record, we will simply display out all requests having first 3 letters common with our input. If the number is less than or equal to three only Binary search will be employed.

iii) Using Author's name :

The Associated-file-2 contains the first 3 letters of the first author's surname and the corresponding relative positions. The procedure to retrieve the details of the request based on the first three letters of the first author's surname is exactly similar to the one described for the requestee's name. In case the surname of the first author is not known, it has been still made possible to retrieve the request record by using any known substring in the author field of the record. This however involves searching the entire author field of the request record for pattern matching. Obviously this search will not be very efficient and precise. Therefore it is advisable that the request be retrieved by using the surname of the first author only.

iv) By using any substring

In case we do not know the exact request number, requestee's name or the author's name, it is possible to retrieve the request record by using any known substring consisting of minimum 3 characters in the record. In this case the user can specify the starting position and ending position in each request record where the search is to be carried out. See Appendix-B for the starting and ending positions of various fields. The entire request file will be searched sequentially in the above specified fields. If the string exists, the record will be retrieved. The substring may be part of the title, publisher code, author name etc. If it is part of title, we may give the starting and ending positions of the title-name field to make the search faster. Though this may not be very efficient retrieval, it will be of immense value when no information regarding the request is available accurately. The precision of retrieval will depend on how unique is the given substring.

- v) By using combination of Request-code, stage, OSW and Publisher-code.

This program helps us in filtering out all those requests which are of interest to us. We can input any, or all of the following combinations :

1. Code-based
2. Stage-wise
3. OSW-based
4. Publisher-code-based

The system will ask for the desired data-fields and using our input, it will retrieve all those requests.

The above queries are answered by running a MIC file called QBPS.MISC.

Before responding to these queries, this program also presents to the user the statistics of the BPS system such as total number of personal, official,

student grant etc. requests, total number of various documents sent, total requests in the system etc. Significantly the statistics also indicate the total number of requests due for generation of various types of documents. This helps the user to generate these documents in time.

2. Generation of Documents in BPS

For accelerating the processing of the requests, the BPS generates various types of documents, at different stages of requests as described in chapter-1. We will describe here, the procedure to generate these documents. BPS generates the following documents.

1. Query letter
2. Indian order letter
3. Foreign order letter
4. Reminder letters (Reminders 1 & 2)
5. Intimation letter to Requestee
6. Miscellaneous letters (order cancellation letters, Advance Payment Letters etc. etc.)

i. Generation of Query letter

a) Type the command

```
.RUN QRYTTY
```

to get the query letter on TTY (terminal) just for glancing to detect some obvious discrepancies.

b) Type the command

```
.RUN QRYLPT
```

to get the query letter on LPT (Line Printer)

It gives one copy for each vendor in addition to two extra copies for library record.

ii) Generation of Indian Order Letter

a) Type the command

```
.RUN INDTTY
```

to get order letter for Indian vendors on TTY.

b) Type the command

```
.RUN INDLPT
```

to get Indian order letters on LPT. This generates one extra copy for each of the order letters, for library record.

iii) Generation of Foreign order letters

a) Type the command

```
.RUN FRNTTY
```

to get order letters to foreign vendors/publishers on TTY.

b) Type the command

```
.RUN FRNLPT
```

to get order letters to foreign vendors/publishers on LPT, with one extra copy for library record.

iv) Generation of Reminder Letters

a) Type the command

```
.RUN REMTTY
```

to get reminder letters to foreign vendors/publishers on TTY.

b) Type the command

```
.RUN REMLPT
```

to get reminder letters to foreign vendors/publishers on LPT, with one extra copy for library record.

It may be noted that for the reminder letters to be sent, the number of stipulated days to have elapsed will differ from country to country. No reminder letters are sent to Indian vendors.

v) Generation of Intimation Letters

We have seen that the four documents described earlier are generated as and when they are asked for. In contrast, the intimation letter is generated automatically as soon as the request reaches stage 8.

In this case a letter is generated directly on printer with no copy for library record, and a flag is set in the request record to indicate that the Intimation letter has been generated.

vi) Miscellaneous Letters

Besides the important documents, we may occasionally require to write miscellaneous types of letters to vendors and publishers. This may include cancellation of orders, providing more information about a book, returning a mistakenly sent book to vendor/publisher, advance payment letters etc. For generating these letters quickly the BPS system contains a module called ANYDOC. This module accepts two files as data files.

The first file TOP.DAT will form the top text of the letter excluding the letter-head. (The letter-head and date will be generated automatically). This will be followed by inputting the name and address of the party and relative position of the records in the Request file) to be included in the document. If there is a range of records to be included, we can specify the starting and ending relative positions separated by colon (e.g. 129:236). Different relative positions should be separated by comma or semicolon (e.g. 250, 252, 276:278). We will process the string comprising of the relative positions and retrieve all needed request records. Using the subroutine PRINTER.FOR, we will print the serial number, author name, title and publisher of all these requests, after splitting them meaningfully. In the end, there will be a bottom-text in some letters. If the bottom-text exists, it will be taken from BOT.DAT.

It may be worth mentioning that for all types of letters (except Intimation letter), the mailing labels are generated automatically for quick despatch.

CHAPTER - 5GENERAL REMARKS

The life span and usefulness of a system depends on the correctness with which it performs and the ease with which users can interact with it.

1) Backup copies of Data Files:

To avoid incorrect updates and corruption of data files, we should keep multiple copies of data file. If during the generation of a document, the computer stops working due to power failure or a hardware trouble, the data files may get partially corrupted. In such cases, we will have to restore the original data files before being updated. After each correct update of the data file, we ensure that at least one backup copy of the same exists.

2) Disk-copy of Hard-copies of documents printed :

Provisions have been made to make a Disk-copy of the document simultaneously with the generation of the document on printer. While preparing the hard-copy of a document, if any unforeseen printer troubles occurs, we will have to re-generate the document again after modifying each and every request included in the document. This will pose a problem when there are lots of requests to be modified. To avoid this we take a Disk-copy of the document along with the hard copy. If the hard-copy is damaged or spoiled, we can use the disk-copy to have another printout of the

same document. The Disk-copy exists only for the last document generated. This Disk-copy can be saved on tape for future references, if necessary.

Impact of BPS

Procuring books for scientific research from abroad by surface mail is a time consuming and laborious process. The scientists are always eager to know what has happened to the books requested by them. As mentioned earlier in the BPS the book request has been split up into 13 distinct stages. Since the research staff can have access to the database, they can use their terminals and find out at which stage the book ordered by them is.

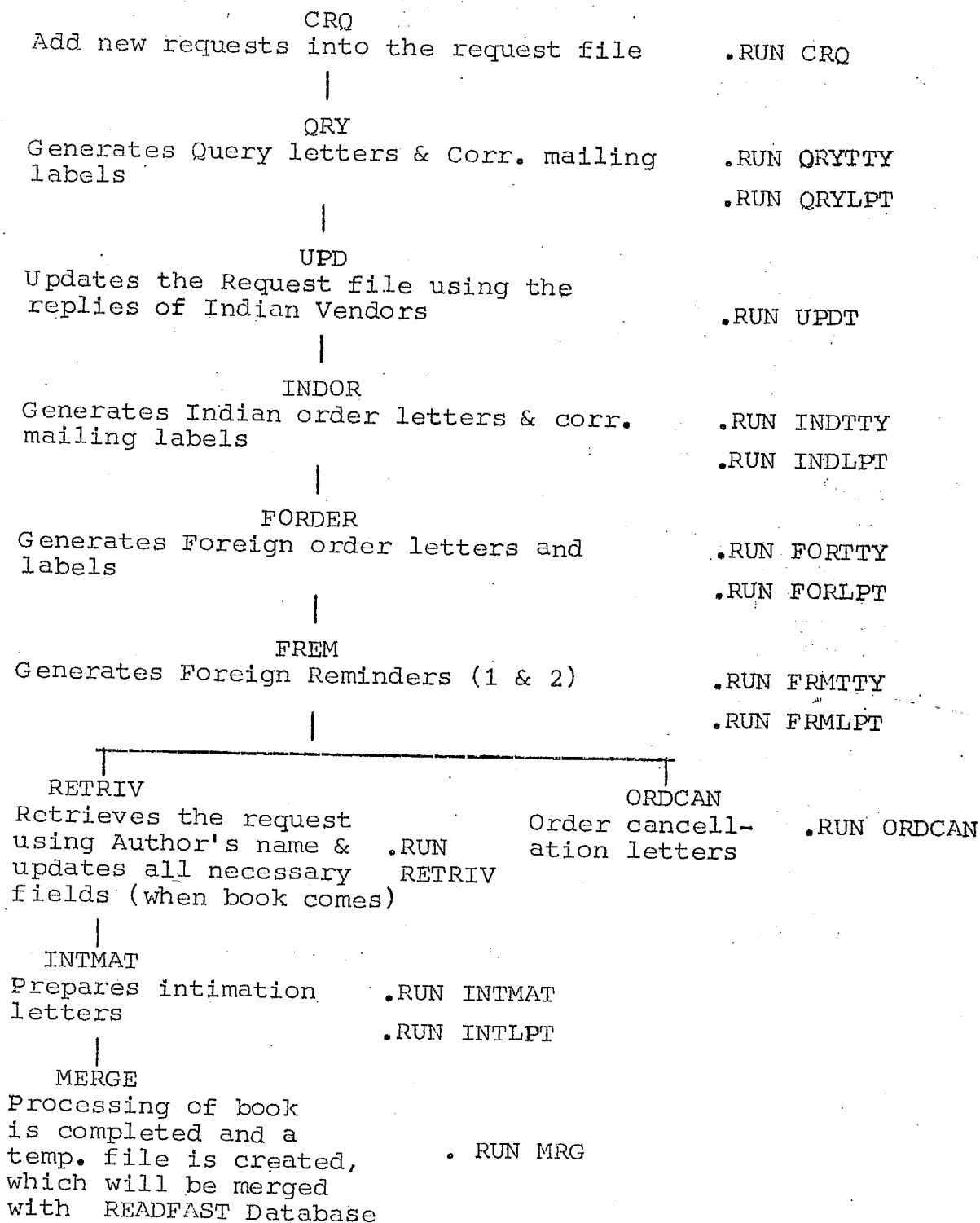
The system of course does all the follow up work such as keeping tracks of books not received, sending reminders etc. This frees the library staff from a large amount of clerical work, for after the book is procured, it even prints out an intimation letter to be given to the requestee. The accession number can also be automatically given, and if essential; it can prepare the Accession Ledger.

The tremendous advantage of this system, is that it gives very vital statistics such as Vendor performance analysis, amount spent for each area, it helps with the budget control etc.

The system also gives various kinds of very useful reports such as all the books ordered by one area, or by a particular author, or all books supplied by a particular

REFERENCES

1. Book Procurement System - Systems Analysis
- D.R. Kulkarni, R.R. Bharucha, S. Bakre
(P.R.L. Technical Note TN-81-08)
2. Manual of Library Management
(A P.R.L. Publication)
3. A class of sorting Algorithms based on Quick-sort
- Roger L. Wainwright
(Communications of ACM, vol.28, no.4)
4. Computer Sorting Techniques
- M.K.Roy and D. Ghosh . Dastidar
5. A fast string searching Algorithm
- Robert S. Boyer, J. Strother Moore
(Communications of ACM, vol.20, no.10)
6. The Periodical Management System
- D.R. Kulkarni, R.R.Bharucha, U.A.Ghiya
(P.R.L. Technical Note TN-80-02)
7. FORTRAN Reference Manual of DEC-System-10

IMPLEMENTATION OF BPS

APPENDIX-B

SUBROUTINE CALLING SEQUENCES

1. CRQ.FOR
 - a. TESTER
 - b. BINSER
 - c. ASORT
 - d. READPB
 - e. SPLIT

2. INDOR.FOR
 - a. READPB
 - b. READRQ
 - c. WRITER
 - d. READER
 - e. PRINTR
 - f. SPLIT

3. QRY.FOR
Same routines as above

4. FORDER
 - a. READPB
 - b. READRQ
 - c. WRITER
 - d. READER
 - e. ASORT
 - f. BINSER
 - g. PRINTR
 - h. SPLIT

5. REMINDER (FREM.FOR)
 - a. READPB
 - b. READRQ
 - c. WRITER
 - d. READER
 - e. ASORT
 - f. BINSER
 - g. DDMMYY - This finds out total number of elapsed
days between any date DD/MM/YY and TODAY
 - h. PRINTER
 - i. SPLIT

APPENDIX-CSTARTING AND ENDING POSITIONS OF VARIOUS FIELDS IN
REQUEST RECORD

<u>Field name</u>	<u>Width</u>	<u>Starting position</u>	<u>Ending position</u>
1. CODE	1	1	1
2. Request number	8	2	9
3. Title	125	10	134
4. Publisher Code	8	135	142
5. Author	40	143	182
6. Edition	6	183	188
7. Copies	1	189	189
8. Stage	4	190	193
9. Price	8	194	201
10. Update Day	4	202	205
11. Update Month	4	206	209
12. Update Year	4	210	213
13. Handling Charge	8	214	221
14. Vendor-Code-1	4	222	225
15. Vendor-Code-2	4	226	229
16. Vendor-Code-3	4	230	233
17. Requestee	20	234	253
18. Section	5	254	258
19. Order Status Word (OSW)	4	259	268
20. Query number	4	263	267
21. Order number	4	267	270
22. Accession number	4	271	274
23. Call number	10	275	284
24. Advance	8	285	292
25. Reminders sent	4	293	296
26. Order Date	6	297	302
27. Order Indicator	1	303	303

APPENDIX-D

MIC COMMAND-FILES

- i) For creating Request File (CRQ.MIC)
- ```
= COMP CRQ.FOR [,,CRQ] /COMP + TESTER.FOR +
BINARY.FOR + CHAIN.FOR + CHRATR.FOR +
MODULE.FOR + READPB.FOR + LUNA.FOR + BSORT.FOR + TRIAL.
TRIAL.FOR + READ50.FOR + ASORT.FOR + DSPLAY.FOR +
REFLCT!.FOR + COLIND.FOR (INCLUDE)

.LOAD

.SAVE CRQ

.RUN CRQ
```
- ii) For Generating Query Letter (QRY.MIC)
- ```
=COMP QRY.FOR [,, QRY] /COMP + LUNA.FOR +
BINARY.FOR + VENPRI.FOR/COP + ASORT.FOR/COMP +
+ PRINTR.FOR/COMP('H)

.LOAD

.SAVE QRYLPT

.RUN
```
- iii) For Generating Mailing Lables for Indian Vendors (MAIL.MIC)
- ```
.COMP MAILAB.FOR/COMP('A)

.LOAD

.SAVE MAIL

.RUN
```

- iv) For updating Request-File using vendor replies  
for our Query letter (UPD.MIC)

```

 COMP
=COMP UPD.FOR/+ LOCK.FOR + CHAIN.FOR + REFLCT.FOR
+ BINRY.FOR + MODULE.FOR + DSPLAY.FOR + SPACE.FOR +
CLSCRN.FOR + LUNA.FOR + MIRROR.FOR + SRCH.FOR +
SERCH.FOR + BSQRT.FOR + READ50.FOR + EMPIRE.FOR +
EXPSCT.FOR + CHRATR.FOR (INCLUDE)

```

```
.LOAD
```

```
.SAV UPDT
```

```
.RU UPDT
```

- v) For Indian Order letter (INDOR.MIC)

```

 COMP
=COMP INDOR.FOR/+ KODAK.FOR + VENPRI.FOR + PRINTR.FOR
+ ASORT.FOR + BINRY.FOR + LUNA.FOR + READER.FOR +
WRITER.FOR + READPB.FOR + READ50.FOR ('A)

```

```
.LOAD
```

```
.SAV INDTTY
```

```
!SAV INDLPT if INCLUDE switch is specified
```

```
. RUN
```

- vi) For Foreign Order letter (FORDER.MIC)

```

 COMP
=COMP FORDER.FOR/+ VENPRI.FOR + READPB.FOR +
READER.FOR + WRITER.FOR + KODAK.FOR + PRINTR.FOR
+ ASORT.FOR + BINRY.FOR + LUNA.FOR + PUBPRN.FOR
+ DIM.FOR + COLIND.FOR + FMAILA.FOR + SMAILA.FOR ('A)

```

```
.LOAD
```

```
.SAV FORTTY
```

```
!SAV FORLPT if INCLUDE switch is specified
```

- vii) For Foreign Reminder letter (FREM.MIC)  
 COMP  
 =COMP FREM.FOR/+ PRINTR.FOR + CHAIN.FOR +  
 ASORT.FOR + BINRY.FOR + PUBPRN.FOR + LUNA.FOR +  
 KODAK.FOR + READPB.FOR + READER.FOR + WRITER.FOR('A)  
 .LOAD  
 .SAVE FRMTTY  
 !SAVE FRMLPT if INCLUDE switch is specified  
 .RUN
- viii) For user Queries to retrieve a Request with  
 Request number, Requestee's name or Author's  
 name (QBPS.MIC)  
 =COMP LOCK.FOR + CHAIN.FOR + KODAK.FOR + DISPLAY.FOR  
 + SRCH.FOR + ISORT.FOR + BSORT.FOR + REFLCT.FOR +  
 BINRY.FOR + MODULE.FOR + CURSOR.FOR + LUNA.FOR +  
 EXPSCT.FOR + JOVIAL.FOR + CHRATR.FOR + READ50.FOR  
 + SPACE.FOR + EMPIRE.FOR + SERCH.FOR  
 .LOAD  
 .SAV QBPS  
 .RUN
- ix) For Retrieving and updating any Request record  
 with Author's name or part of the title (RETRIV.MIC)  
 COMP  
 = COMP RETRIV.FOR/+ EMPIRE.FOR + RQSTAT.FOR +  
 LUNA.FOR + EXPSCT.FOR + COLIND.FOR + BSORT.FOR  
 + MODIFY.FOR + ADRPRN.FOR + CHRATR.FOR + CLSCRN.FOR  
 + READ50.FOR + BINRY.FOR + DSPLAY.FOR + REFLCT.FOR  
 + SERCH.FOR (INCLUDE)  
 .LOAD  
 .SAV RETRIV



x) For modifying various fields in Request record

(MODIFY.MIC)

COMP

=COMP MODIFY.FOR/+ CHAIN.FOR + BINRY.FOR +  
MODULE.FOR + READ50.FOR + UNSTRN.FOR +  
COMPRS.FOR + CHRATR.FOR + ASORT.FOR (INCLUDE)

.LOAD

.SAV MODIFY

.RUN

xi) For listing all requests satisfying a  
combination of conditions (COMB.MIC)

COMP

=COMP TIPTOP.FOR/+ CHAIN.FOR + PRINTR.FOR + BINRY.FOR  
+ ASORT.FOR + DSPLAY.FOR + LUNA.FOR + READER.FOR  
+ WRITER.FOR + READPB.FOR + TRIAL.FOR + ONEBYI.FOR  
+ CHRATR.FOR + READ50.FOR ('A)

.LOAD

.SAV COBLPT

.RUN

xii) To Generate any desired document (ANYDOC.MIC)

!At first generate the top and bottom text files using

! EX CRETXT.FOR

COMP

=COMP STAKR.FOR/+ LUNA.FOR + PRINTR.FOR +

ASORT.FOR + BINRY.FOR + COLIND.FOR + VENPRI.FOR +

READPB.FOR + READER.FOR + WRITER.FOR + SPACE.FOR('A)

.LOAD

.SAV ANYLPT

.RUN

xiii) To Generate Intimation letter (INTMAT.MIC)

```
=COMP INTMAT.FOR + PRNTEXT.FOR + READER.FOR +
WRITER.FOR + PRINTR.FOR + EXPSCT.FOR + VENPRI.FOR
+ BINRY.FOR + LUNA.FOR + ASORT.FOR + READPB.FOR
+ READ50.FOR ('A)
```

```
.LOAD
```

```
.SAVE INTMAT
```

```
.RU
```

xiv) For creating Vendor file (CREVEN.MIC)

```
EX CREVEN.FOR + COLIND.FOR + LUNA.FOR
```

```
.LOAD
```

```
.SAVE CREVEN
```

xv) For Mailing Labels for Foreign Publishers(FMAIL.MIC)

```
.COMP FMAIL.FOR + ASORT.FOR + BINRY.FOR ('A)
```

```
.LOAD
```

```
.SAVE FMAIL
```

xvi) To Refresh the Request File by deleting all

unnecessary requests (CODE = '\*')

```
COMP
=COMP REFRSH.FOR/+ LUNA.FOR + PRINTR.FOR +
```

```
ASORT.FOR + BINARY.FOR + COLIND.FOR + VENPRI.FOR ('A)
```

```
.LOAD
```

```
.SAV REFRSH
```

xvii) To modify the Publisher File (MODPUB.MIC)

```
COMP
=COMP MODPUB.FOR/+ BINRY.FOR + ASORT.FOR +
```

```
NOTALF.FOR + COLIND.FOR + LNOTNM.FOR ('A)
```

```
.LOAD
```

xviii) To list all requests in a particular stage  
(ISTAGE.MIC)

```
=COMP ISTAGE.FOR + READER.FOR + WRITER.FOR +
READPB.FOR + PRINTER.FOR + LUNA.FOR + BINARY.FOR
+ ASORT.FOR + READ 0.FOR
.LOAD
.SAVE W : ISTAGE
```

xix) To search any arbitrary pattern in Request records  
(SERCH.MIC)

```
 COMP
=COMP SERPAT.FOR + SRCH.FOR + SERCH.FOR/+ LUNA.FOR
+ COLIND.FOR + CHARTR.FOR + CHAIN.FOR + DSPLAY.FOR
+ REFLECT.FOR + READ50.FOR
. LOAD
. SAVE SERCH
```

xx) To Restor the Associated files (NEM.REP, NRQ.REP,  
NEM.RQN) from the Request file (RESTOR.MIC)

```
.COMP RESTOR.FOR/COMP
.LOAD
.SAVE RESTOR
```