



US 20050038830A1

(19) **United States**

(12) **Patent Application Publication**
Lee

(10) **Pub. No.: US 2005/0038830 A1**

(43) **Pub. Date: Feb. 17, 2005**

(54) **DISTRIBUTION OF MAINFRAME DATA IN THE PC ENVIRONMENT**

(52) **U.S. Cl. 707/200**

(76) **Inventor: Hyung Sup Lee, Bayside, NY (US)**

(57) **ABSTRACT**

Correspondence Address:
LADAS & PARRY
26 WEST 61ST STREET
NEW YORK, NY 10023 (US)

A method of distributing mainframe software and data using PC-based data media is disclosed, comprising a mainframe program for converting a mainframe sequential dataset or all or part of a mainframe library of texts or load modules to a sequential dataset of fixed length records and for reverting these converted records to its original form in the same or another mainframe data center, and procedures for downloading the converted mainframe data to a PC and uploading the PC file to a mainframe. The converted mainframe data downloaded to a PC can be delivered to another mainframe data center using floppy diskettes, recordable CD-ROM, Internet Website, TCP/IP FTP, or email attachment, as an alternative to magnetic tapes. The PC hard disk is used as a backup storage of the mainframe libraries, and a mainframe text library downloaded to a PC can be edited on a local or remote stand-alone PC.

(21) **Appl. No.: 10/935,464**

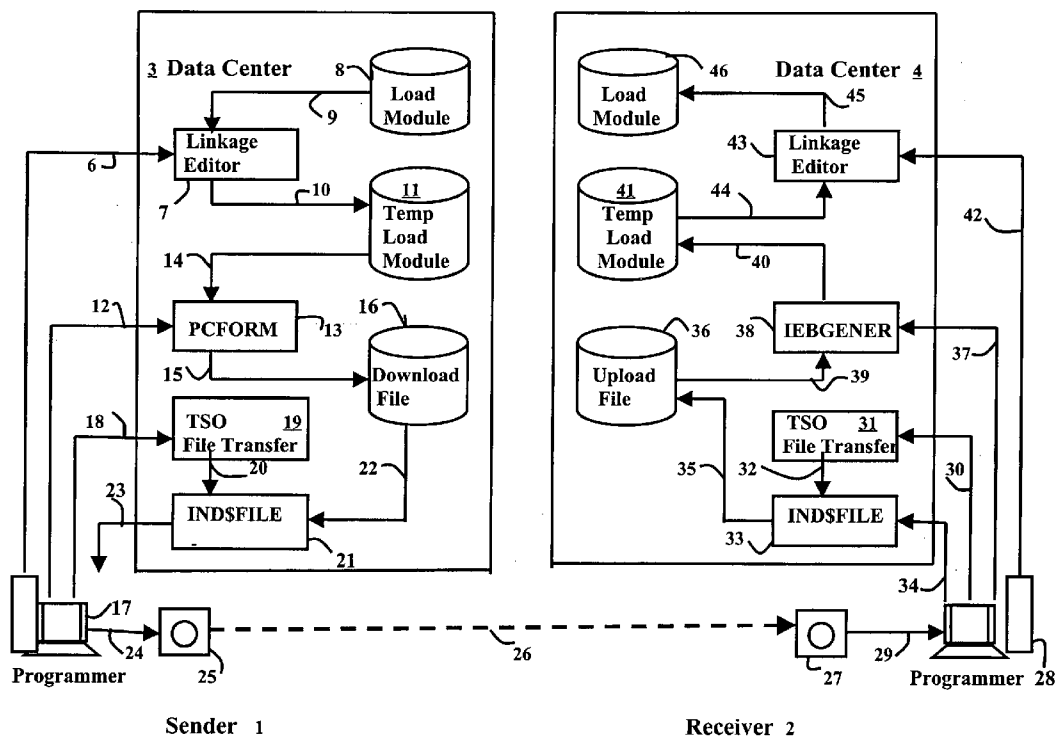
(22) **Filed: Sep. 7, 2004**

Related U.S. Application Data

(63) **Continuation of application No. 09/726,462, filed on Nov. 29, 2000.**

Publication Classification

(51) **Int. Cl.⁷ G06F 12/00**



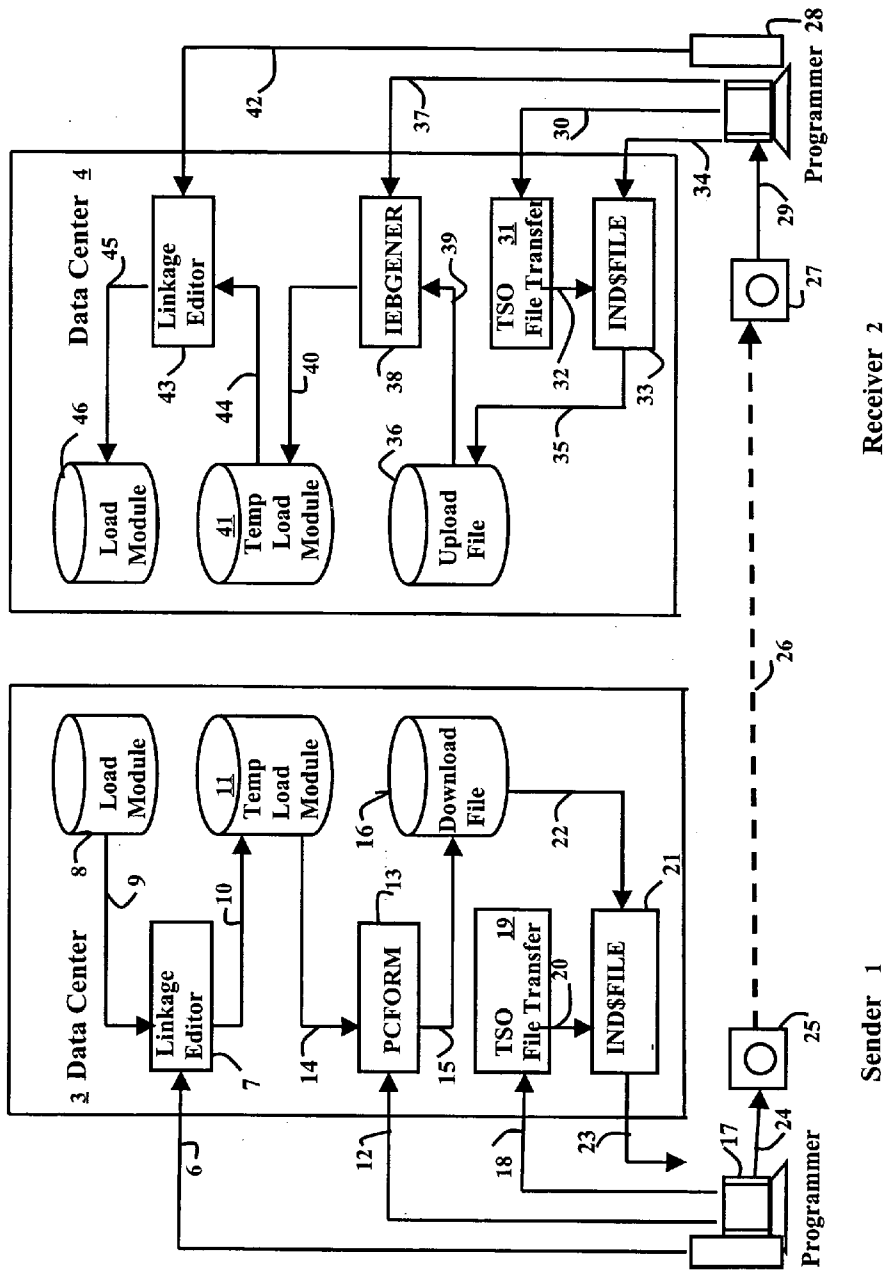


Fig. 1 Transporting a load module when the receiver doesn't have the program PCFORM

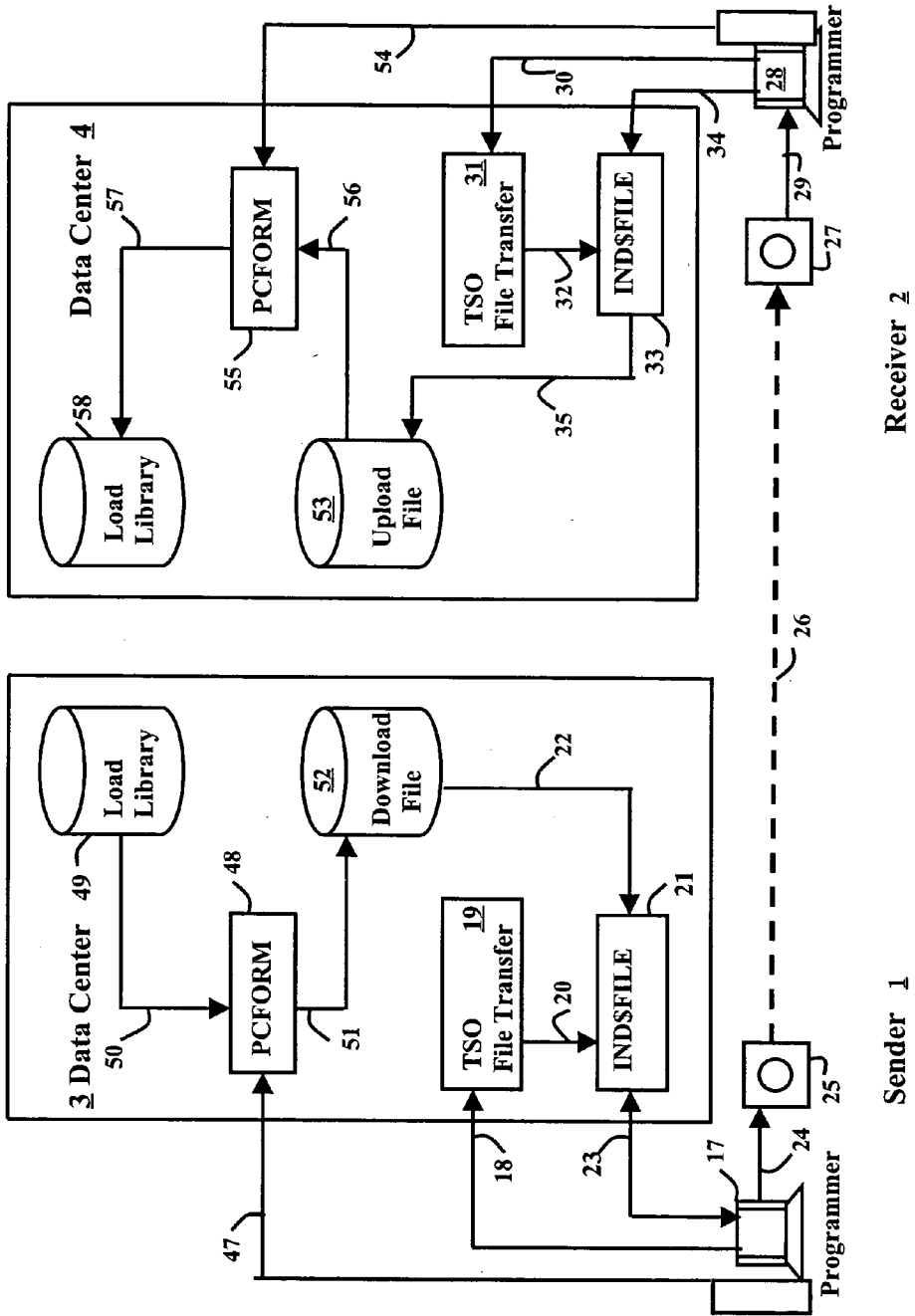


Fig. 2 Transporting a load library when the receiver has the program PCFORM

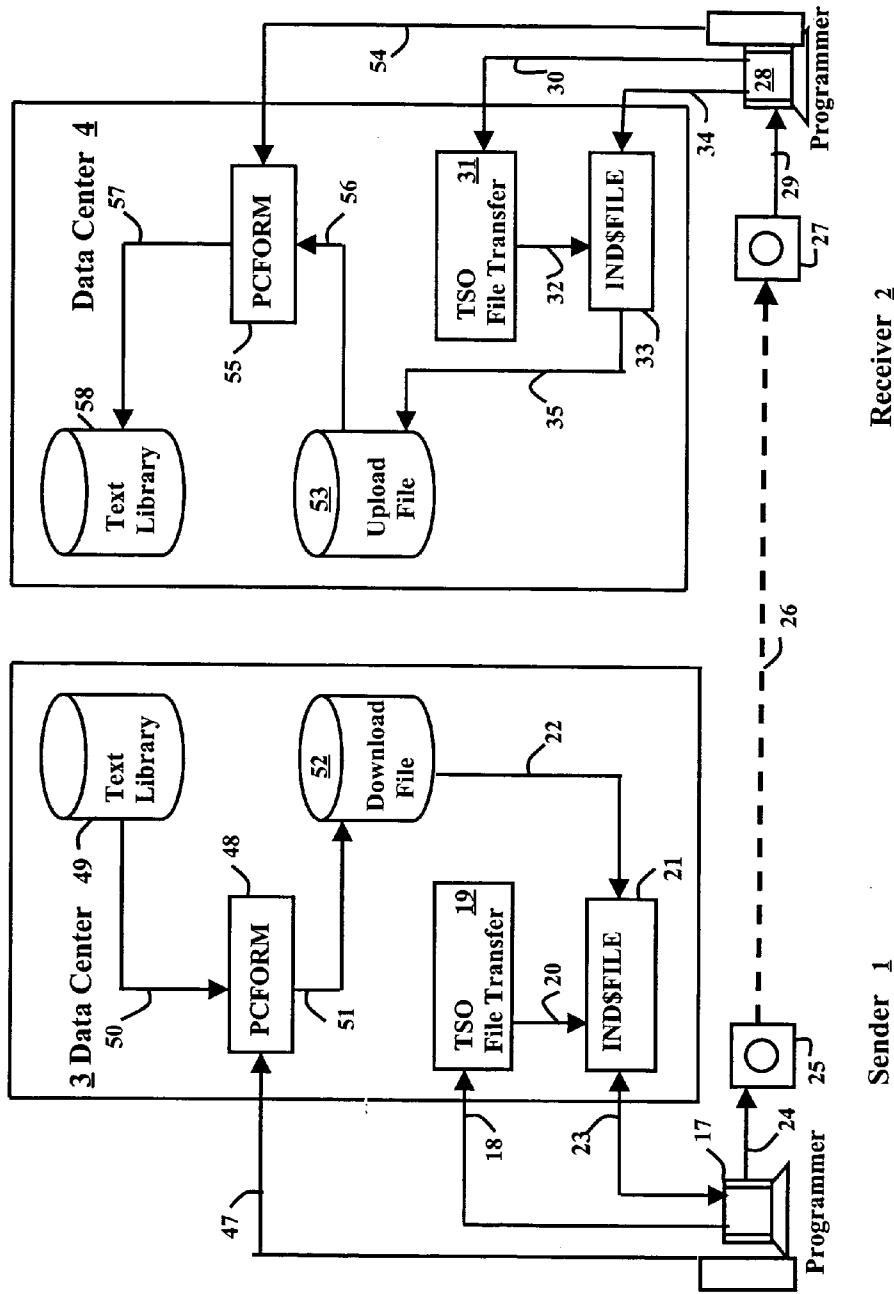


Fig. 3 Transporting a text library when the receiver has the program PCFORM

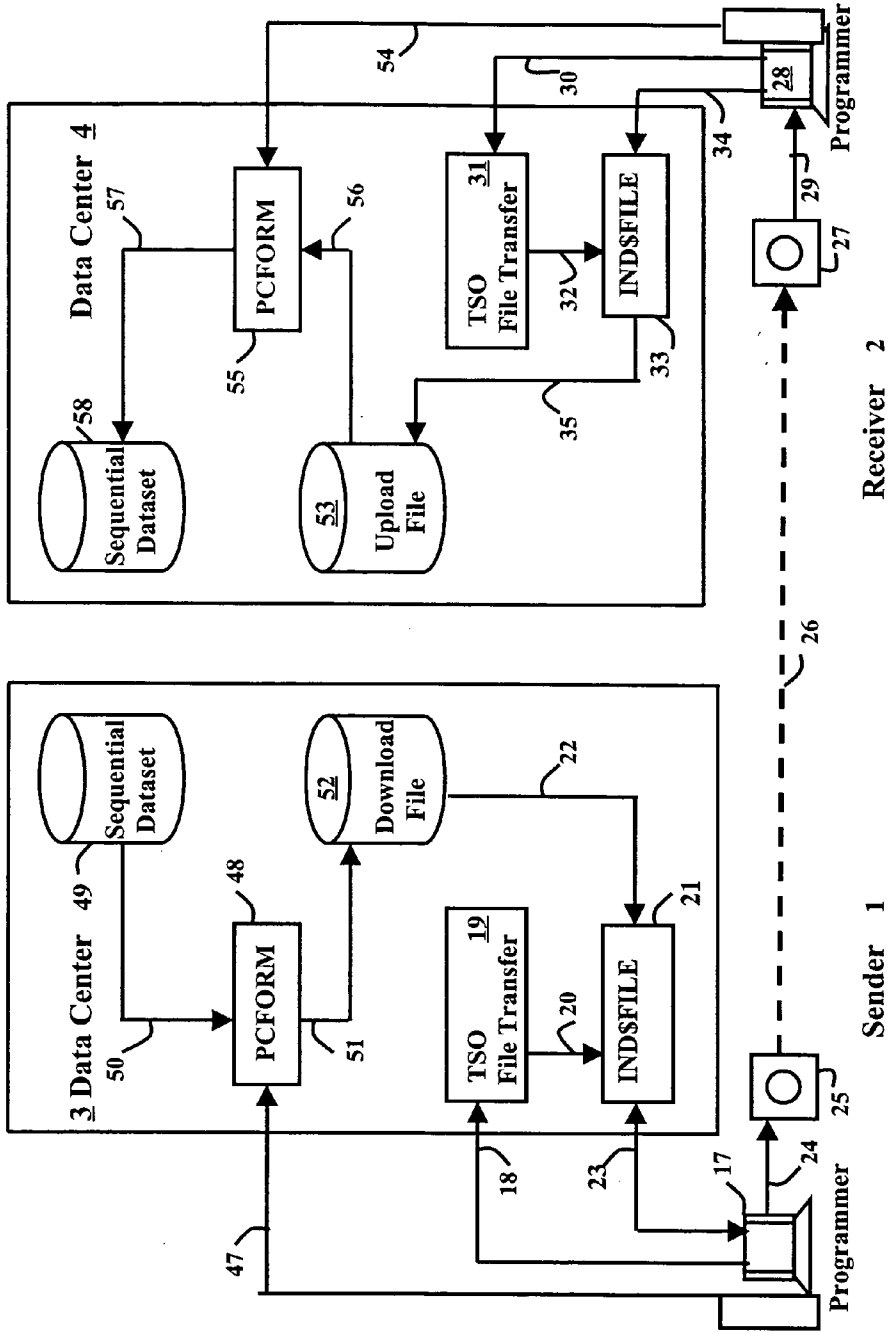


Fig. 4 Transporting a sequential dataset of fixed, undefined or variable record format when the receiver has the program PCFORM

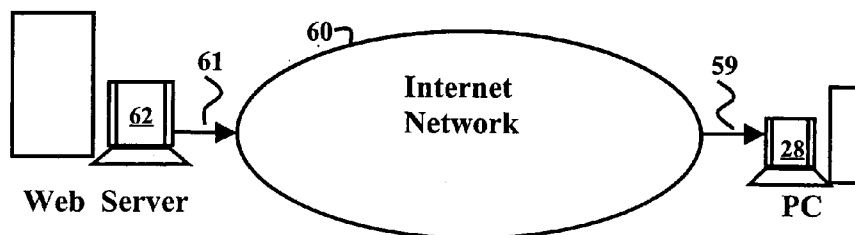


Fig. 5. World Wide Web and the Internet Download

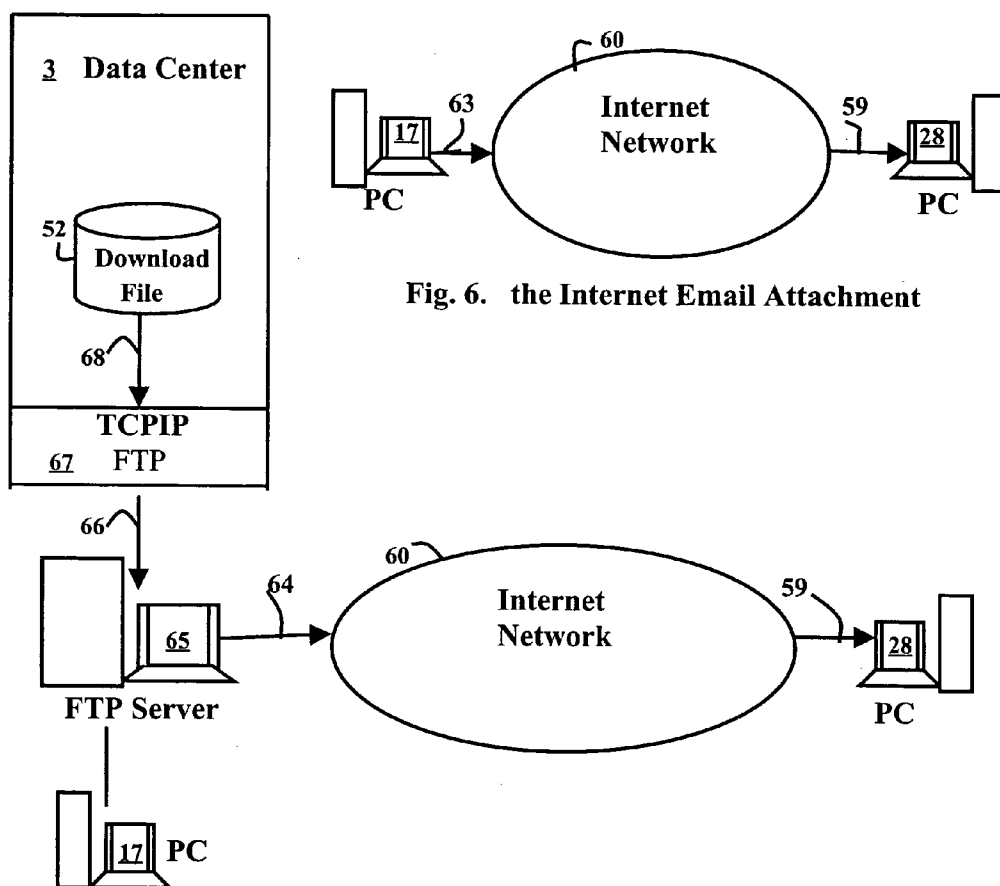


Fig. 6. the Internet Email Attachment

Fig. 7. FTP Download

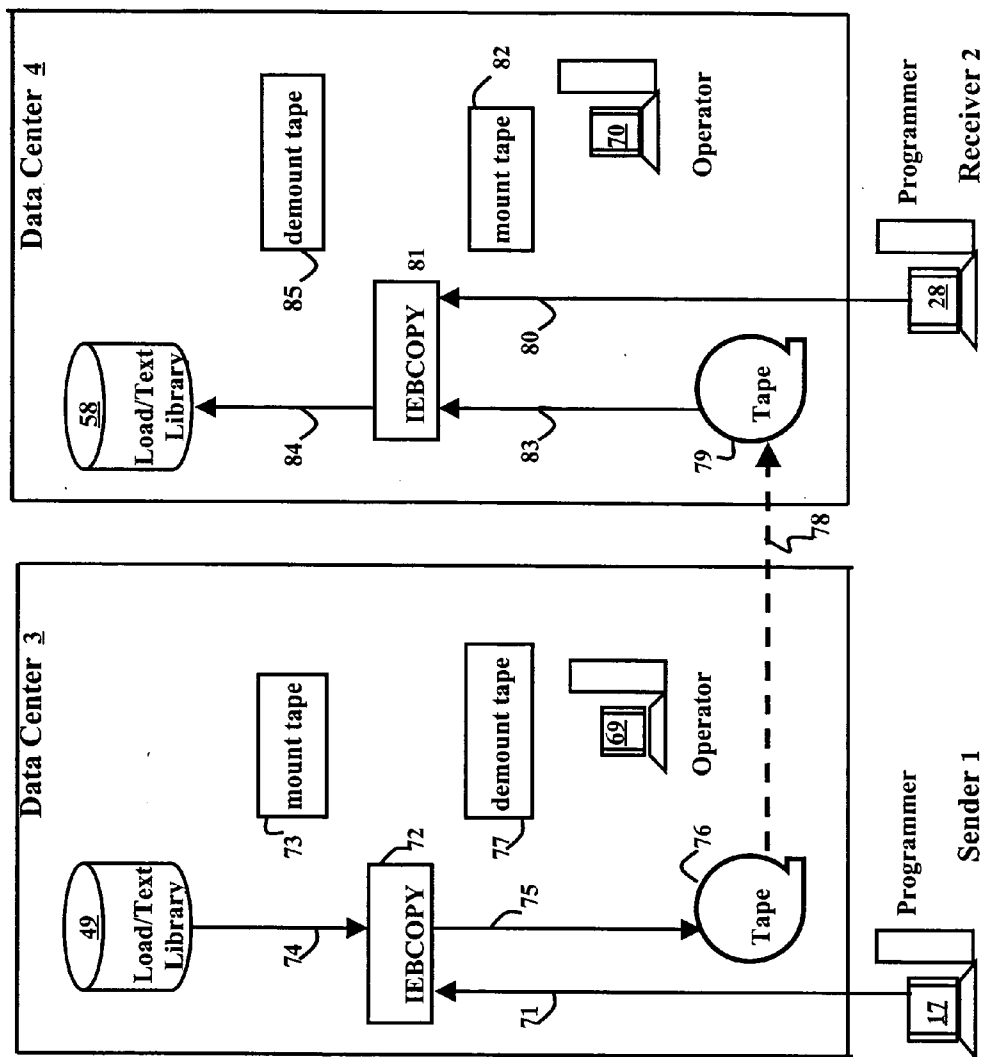


Fig. 8 The Conventional Tape Delivery of a Load/Text Library

Fig. 9. The Conversion Program PCFORM

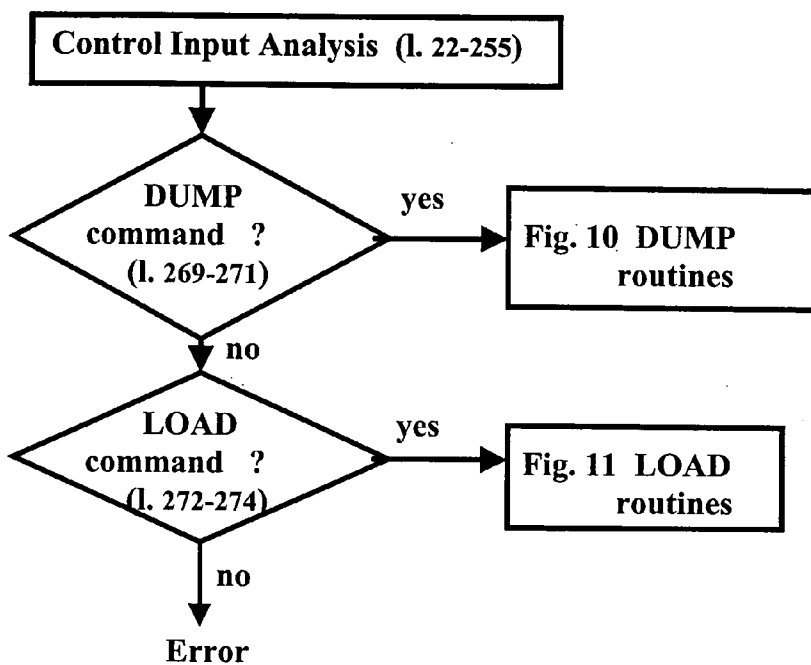


Fig. 10 DUMP Routines

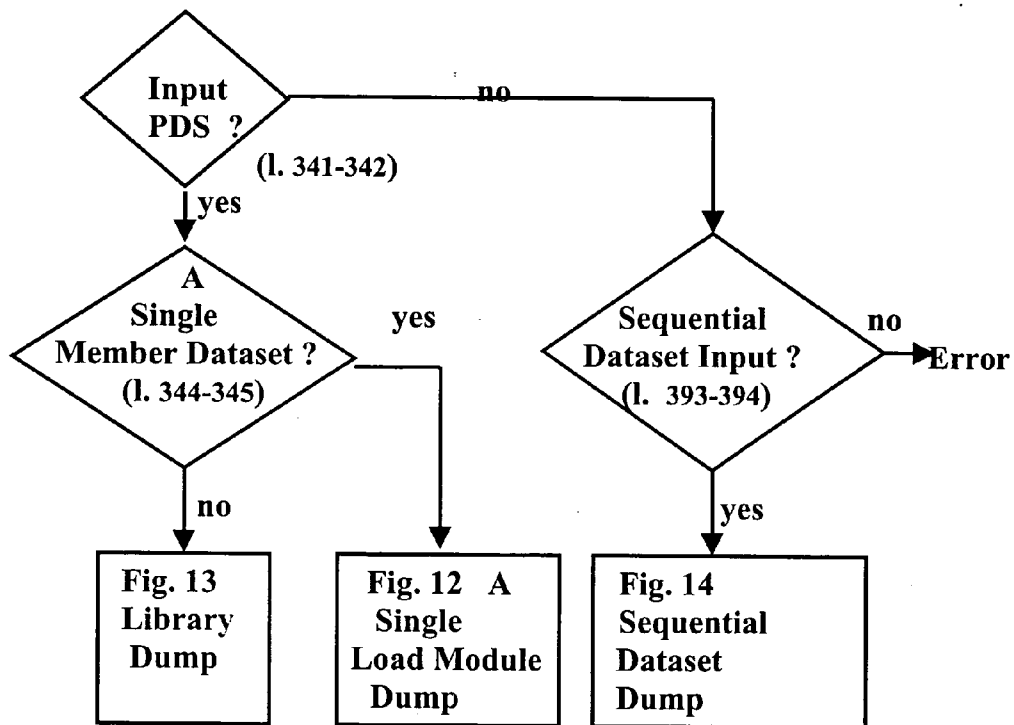


Fig. 11 LOAD Routines

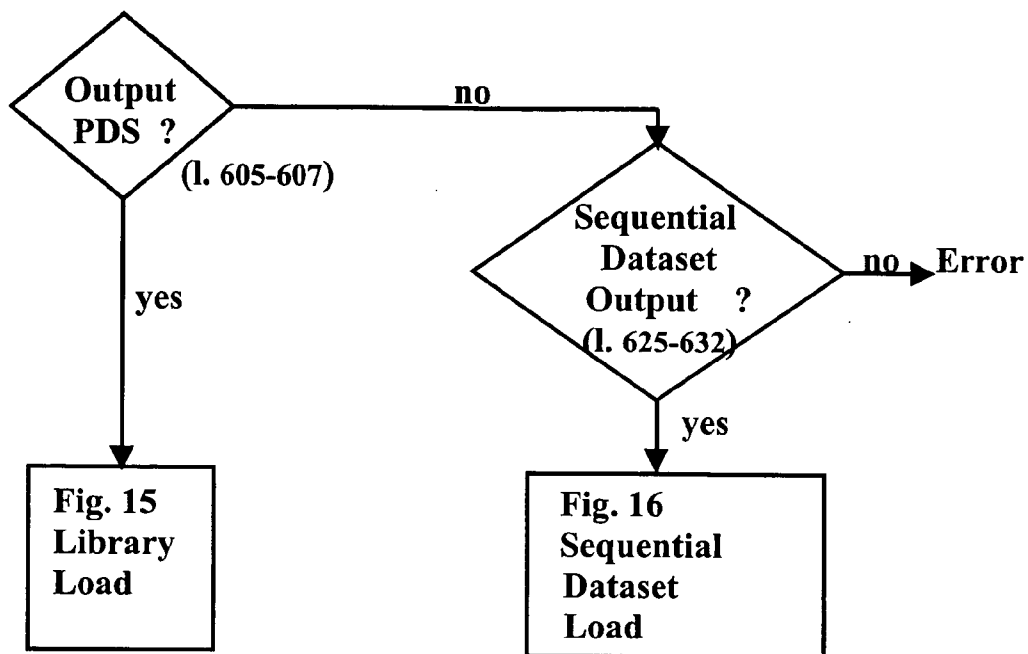


Fig. 12. A Single Load Module Dump Routine

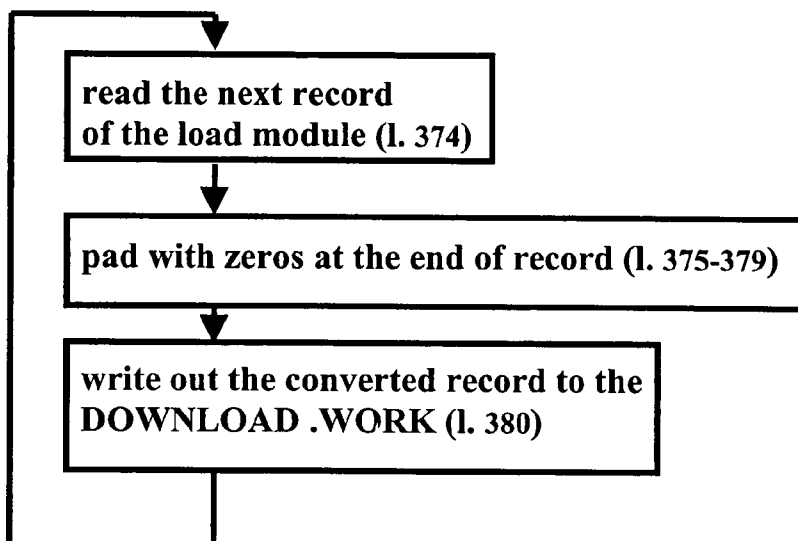


Fig. 13. Library DUMP Routine

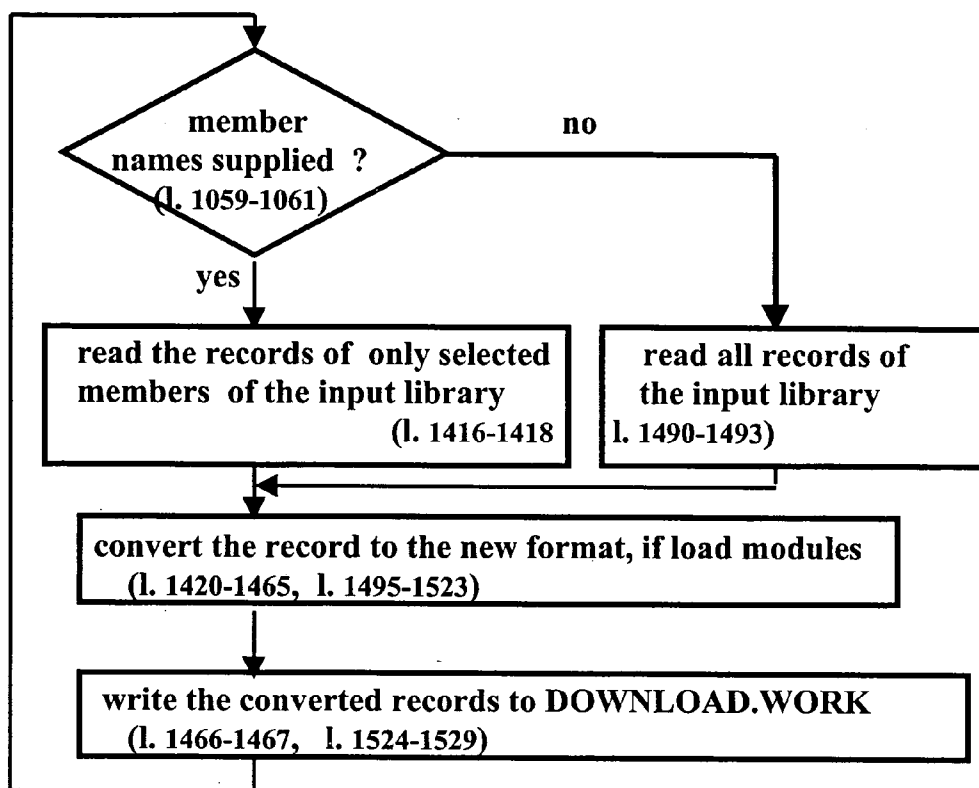


Fig. 14. Sequential Dataset Dump Routine

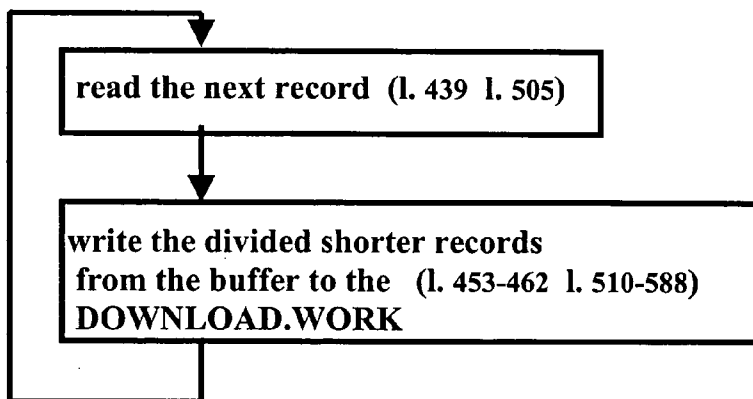


Fig. 15. Library LOAD Routine

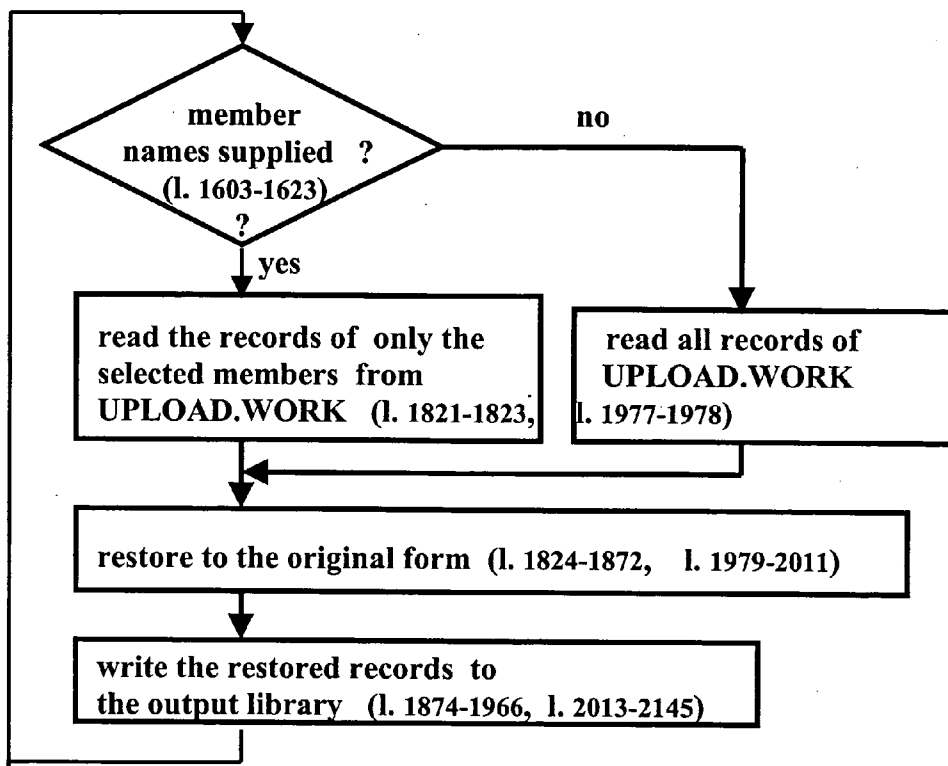


Fig. 16. Sequential Dataset LOAD Routine

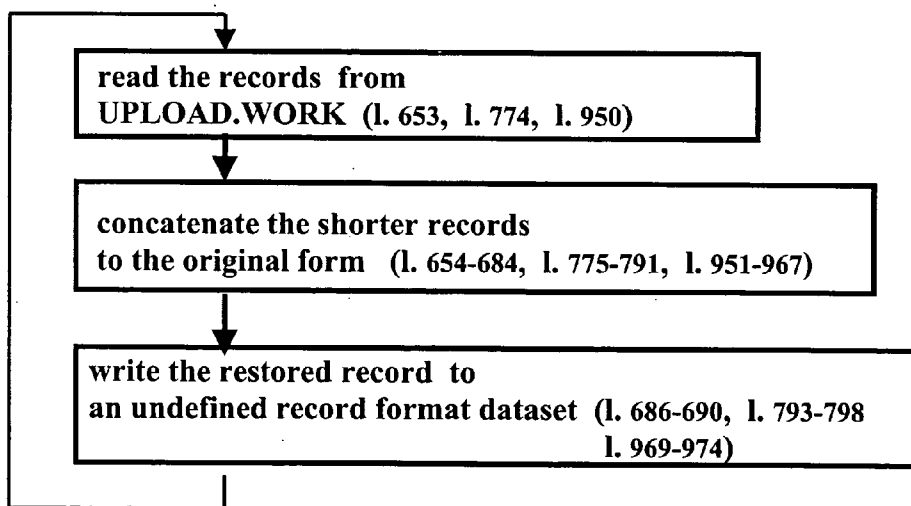
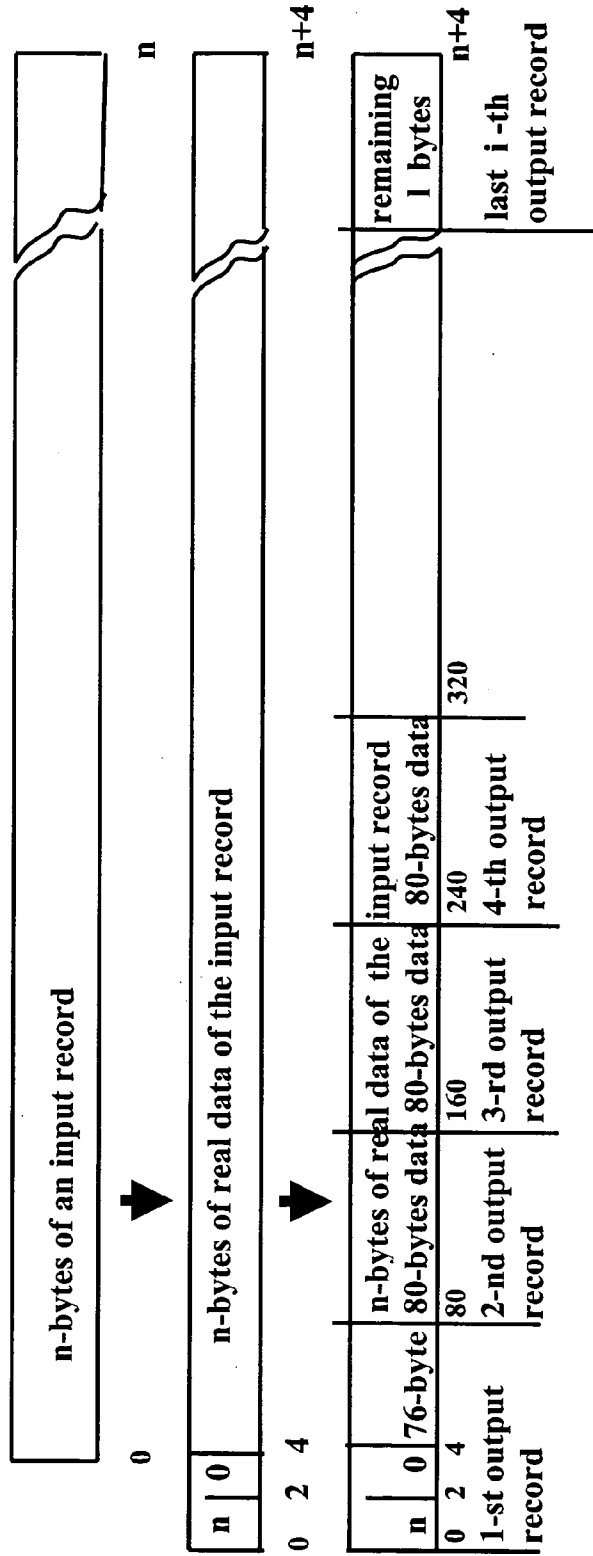


Fig. 17 A Conversion Method of A 4-bytes Header and Division



* The length of the input record is assumed as n bytes long.

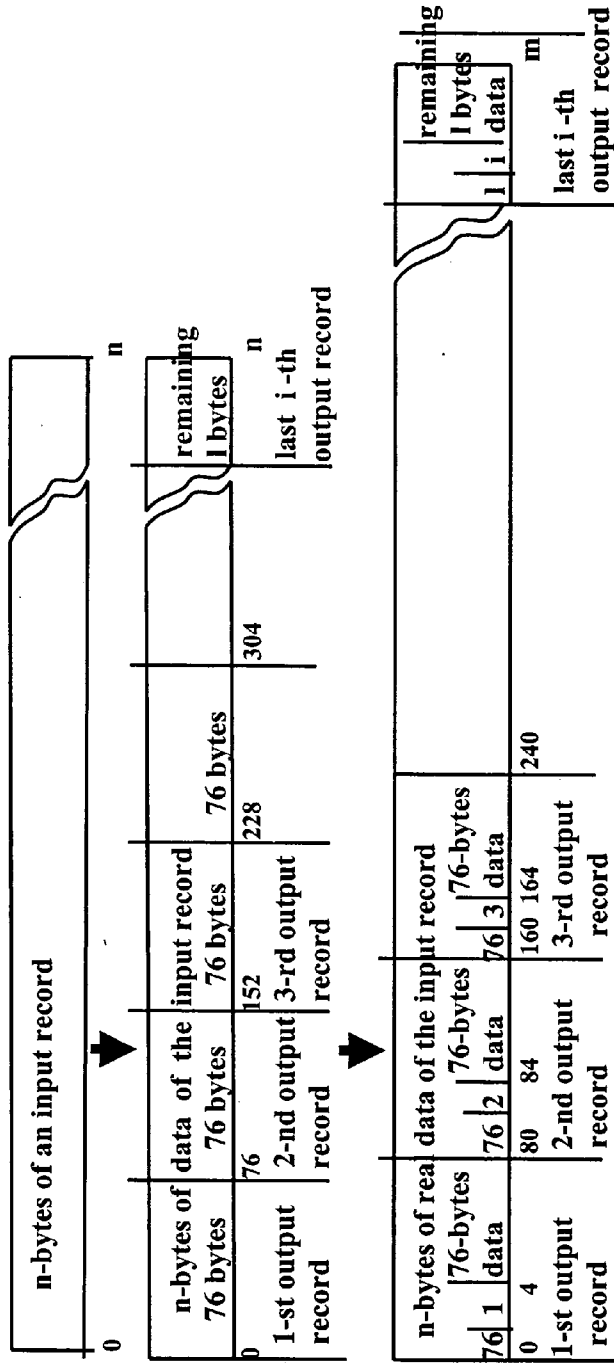
The logical record length of the output (Download & Upload) sequential dataset is assumed as 80-bytes.

A 4-bytes header with 2-bytes of real data length, e.g., n , and 2 null bytes.

$$i = \lfloor (n + 4 + 79) / 80 \rfloor \quad \text{where } i\text{-th record is the last output record of an input record and } \dots \text{ is an integer value}$$

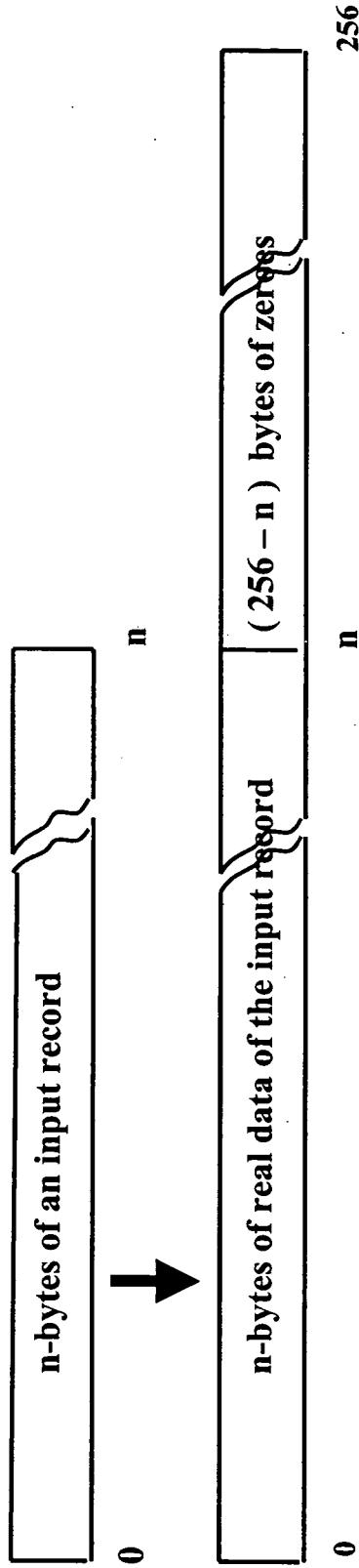
$$l = (n + 4) - ((i - 1) * 80) \quad \text{bytes length of the last output record}$$

Fig. 18 A Conversion Method of Division into Shorter Records and A 4-bytes Header for Each Divided Shorter Record



* The input record is assumed as n bytes long.
 The logical record length of the output (Download & Upload) dataset is assumed as 80-bytes long.
 A 4-bytes header, for each divided shorter record, with
 2-bytes of real data length, e.g., 76 and
 2-bytes of sequence number of divided records
 $i = \lfloor (n + 75) / 76 \rfloor$ where i -th record is the last output record and [...] is an integer value
 $l = n - ((i - 1) * 76)$ bytes of real data in the last output record
 $m = (i - 1) * 80 + (l + 4)$

Fig. 19 Zero Padding of a record of a single load module



* The logical record length of the output (Download & Upload) sequential dataset is the same as the block size (maximum record length) of the input load module library where the target single load module is located. It is assumed as 256-bytes.

The length of the input record is assumed as n bytes long where the maximum value of n is the block size of the library and it is assumed as 256 and the minimum value of n is 1.

DISTRIBUTION OF MAINFRAME DATA IN THE PC ENVIRONMENT

FIELD OF THE INVENTION

[0001] The invention relates to the distribution method of mainframe data using PC-based data media. More specifically, the invention relates to a mainframe program for allowing the file transfer of mainframe sequential datasets and libraries of texts or load modules between a mainframe computer and a PC and between two mainframe computers using floppy diskettes, recordable CD-ROM, Internet world wide web, TCP/IP FTP, and email attachment, thus providing an alternative to magnetic tape reels and tape cartridges, also allowing the PC hard disk to be used as a backup storage of the mainframe libraries, as well as for allowing a mainframe text library which is downloaded to a PC to be edited on a local or remote stand-alone PC workstation.

DISCUSSION OF RELATED ART

[0002] There are currently more than 21,000 mainframe data centers worldwide. More than 2 trillion dollars has been invested in mainframe software (around 150 billion lines of COBOL code alone worldwide). Mainframe computers are becoming less expensive (\$100,000/MIPS in 1992, \$2,300/MIPS in 2000, and \$840 in 2002). 70% of all business information resides in mainframe databases. The operation costs of the distributed computing based on mini-computers are high. More than anything else, businesses need machines with enough computative power, a need which mini-computers cannot satisfy. Mainframes will continue to be the main workhorses for big businesses for the foreseeable future.

[0003] Mainframes have been upgraded by replacing the former “dumb” terminals, such as the IBM 3270s terminals which provided little more than a keyboard and a display screen, with desktop personal computers (PCs). These PCs can also be used as stand-alone computers for text editing, word processing, and other office tasks, when not being used as mainframe terminals.

[0004] A PC can connect to a mainframe system with the proper software and proper communication link. A PC terminal connects to and communicates with a mainframe through an IBM Systems Network Architecture(SNA) 3270 protocol emulation software. The PC can operate as a LAN terminal using a LAN server, an SNA gateway, and a 3270 emulation software, e.g., Attachmate Extra Personal Client™. The PC can work as a stand-alone mainframe terminal through a Synchronous Data Link Control (SDLC) communication board, a leased-line modem, and a 3270 emulation software, such as Attachmate Extra™, 3270 Elite™, or Dynacomm/Elite™. Or an Internet-enabled PC can work as a stand-alone mainframe terminal through a Telnet server which supports TCP/IP application protocol TN3270 clients with TN3270 Telnet SNA 3270 protocol emulation software, such as QWS3270 Plus TN3270 Telnet application™ of Jolly Giant Software or E-Term™ for IBM of DCSi.

[0005] The following procedure (Procedure #1) is a PC-mainframe connection procedure for an Internet-enabled PC with QWS3270 Plus TN3270 application software of Jolly Giant Software:

Procedure #1

[0006] A PC-to-Mainframe Connect Using TN3270 Telnet

- [0007] 1. Start the program QWS3270.

[0008] 2. ‘QWS3270PLUS TN3270 Telnet’ screen comes up.

[0009] Click the Host menu.

[0010] 3. Click Connect from the drop-down list.

[0011] 4. ‘Host Selection’ screen comes up.

[0012] Type Session_Name.e.g., lj01 decided by Host site

[0013] Type Host_Name. e.g., 123.45.67.89 a Telnet server’s IP address

[0014] Type Port number. e.g., 21 or 23. decided by Host site

[0015] Choose Terminal Type. e.g., IBM-3278-2

[0016] Click Connect button.

[0017] 5. Internet Dialing and Connecting screens come up.

[0018] 6. The mainframe data center’s main SignOn screen comes up.

[0019] Type User Id. e.g., a TSO User ID. lj01

[0020] Type Password. e.g., password of ij01

[0021] 7. TSO screen comes up.

[0022] A PC connected to a mainframe with an IBM SNA 3270 protocol emulation software or the TN3270 Telnet software supports file transfers between the mainframe and the PC using the IBM mainframe program “INDSFILE”. An Internet-enabled PC connected to a mainframe FTP server using the TCP/IP file transfer protocol (FTP) also supports the file transfer between the mainframe and the PC.

[0023] The three file transfer procedures (Procedures #2-4) between a mainframe and a PC are explained as examples. The first procedure (Procedure #2) is a file transfer during a TSO session using Attachmate Extra SNA 3270 emulation software:

Procedure #2

[0024] TSO File Transfer With Attachmate Extra Emulator

[0025] 1. Get out of ISPF screen.

[0026] 2. On ‘READY’ TSO prompt screen,

[0027] click Tools menu.

[0028] 3. Click Transfer_File from drop-down list.

[0029] 4. On the ‘Transfer File’ screen:

[0030] Type PC_File name.

[0031] Type Host_File name inside quotes.

[0032] Choose the transfer direction between Send or Receive.

[0033] Choose a Scheme between Binary or Text mode.

[0034] Choose TSO between the Host Systems TSO, CICS, and CMS.

[0035] Click Add_to_List button.

- [0036] Click Transfer button.
 - [0037] 5. 'File Transfer Progressing' screen comes up.
 - [0038] 6. After the transfer is complete,
 - [0039] the 'Transfer Summary' screen appears.
 - [0040] Check for the message 'TRANS03 File Transfer complete.'
 - [0041] 7. Click OK button.
 - [0042] 8. A 'READY' TSO prompt screen will appear.
- [0043] The second procedure (Procedure #3) is for an FTP file transfer started from the Attachmate main screen without a TSO session:

Procedure #3

[0044] FTP File Transfer From Attachmate Main Screen

- [0045] 1. Terminate TSO session.
- [0046] 2. On Attachmate initial session list screen,
- [0047] click Tools menu.
- [0048] 3. Click File_Transfer from drop-down list.
- [0049] 4. On 'File transfer' screen:
- [0050] Type PC_File name.
- [0051] Type Host_File name inside quotes.
- [0052] Choose the transfer direction between Send or Receive.
- [0053] Choose a Scheme between Binary or Text mode.
- [0054] Click Transfer button.
- [0055] 5. On 'FTP Connect' screen:
- [0056] Type Host_IP address. e.g., 123.45.67.89 or alias of FTP server
- [0057] Type User_ID. e.g., a TSO User ID, lj01
- [0058] Type Password. e.g., a TSO User ID's password
- [0059] Click OK button.
- [0060] 6. 'File transfer' screen comes up.
- [0061] 7. On 'Transfer Summary' screen,
- [0062] check 'Transferred successfully' Comment message.
- [0063] Click OK button.
- [0064] 8. Attachmate initial session list screen comes up.

[0065] The third procedure (Procedure #4) is FTP File Transfer started from MS-DOS prompt screen:

Procedure #4

[0066] FTP File Transfer From MS-DOS Screen

- [0067] 1. Minimize or terminate all other applications including Attachmate Extra.

- [0068] 2. Start MS-DOS.
- [0069] 3. On MS-DOS screen with C:\WINDOWS>prompt:
- [0070] Type FTP after C:\WINDOWS>prompt.
- [0071] Press Enter key.
- [0072] 4. FTP> prompt comes up.
- [0073] Type OPEN and Host_IP-address at FTP> prompt. e.g., OPEN 123.45.67.89 or OPEN abc.def.ghij.com.
- [0074] Press Enter key.
- [0075] 5. Internet connection starts,
- [0076] Dialing screen and Connecting screens come up.
- [0077] And finally the Internet screen disappears.
- [0078] 6. Click the minimized FTP program icon.
- [0079] 7. MS-DOS screen appears again with FTP messages of
- [0080] "Connected" and others.
- [0081] Type User ID at User prompt. e.g., lj01
- [0082] Press Enter key.
- [0083] Type password at Password prompt
- [0084] Press Enter key.
- [0085] 8. "Logon successful" FTP message comes up with FTP>prompt.
- [0086] Type BINARY or ASCII at FTP> prompt.
- [0087] At the FTP> prompt,
- [0088] type GET 'LJ01.TEXT.LIB(TEXT1)' C:\LJ01\TEXT1.TXT for download from mainframe to PC or
- [0089] type PUT C:\LJ01\TEXT1.TXT'LJ01.TE\XT.TEMP1' for upload from PC to mainframe.
- [0090] Press Enter key
- [0091] 9. Transfer starts.
- [0092] 10. After completion of transfer,
- [0093] check for '250 Transfer completed successfully' FTP message.
- [0094] 11. Type QUIT at FTP>prompt.
- [0095] Press Enter key.
- [0096] 12. Main Windows screen with all icons comes up.
- [0097] Click the minimized MS-DOS icon.
- [0098] 13. MS-DOS screen comes up again with C:\WINDOWS>prompt.
- [0099] Type EXIT at C:\WINDOWS> prompt.
- [0100] Press Enter key.
- [0101] 14. Original Windows screen with all icons comes up.

[0102] After the mainframe file is transferred to a PC, the mainframe data stored in a PC can be copied to a PC-related data media repeatedly. Ubiquitous PC usage worldwide and the file transfer capability make it possible to send and receive data between mainframe data centers using PCs. Furthermore, the rise of PC-based technologies in exchanging information, such as email attachments using SMTP, the Internet world wide web, and FTP send/receive facility, allow data centers to transfer mainframe data using methods which were not previously available.

[0103] However, mainframe installations still send their mainframe libraries of texts and program load modules to other data centers using the half-inch thick, 9-track, 1600/6250 BPI, 1,200 foot 8.5-inch diameter or 2,400 foot 10.5-inch diameter round reel tapes created from IBM 3420/3430 Magnetic-Tape Units or 18-track, 550 foot 38,000 BPI tape cartridges created from IBM 3480 Magnetic-tape Units located in the mainframe computer rooms. There have been no alternatives until now. These magnetic tapes or cartridges require IBM proprietary magnetic tape units.

[0104] There are two main reasons for using magnetic tapes exclusively to deliver a mainframe library. First, a part or whole of a mainframe library can not be downloaded (file transfer from a mainframe to a PC) as a whole. Currently, each member of the mainframe library can only be downloaded individually one at a time. Therefore a mainframe library can not be delivered to other data centers using PC-based data media. At mainframe data centers, libraries are processed only by dataset utility programs developed by IBM. Currently, IBM does not provide the capacity to distribute a mainframe library via PC-related data media.

[0105] Second, only sequential datasets of fixed length records (a "flat" file, in PC terms) or text datasets of undefined record format can be delivered to another data center using PC-related data media, and be safely reverted to its original form at another mainframe data center. While any sequential datasets can be downloaded to a PC a single file at a time, not all sequential datasets can be reverted to their original form when uploaded (file transfer from a PC to a mainframe) at the same or another data center.

[0106] During the downloading process each mainframe record is concatenated to the previous record to create a long thread of string, and stored upon the PC. The uploading process breaks down this long concatenated string of PC data into separate mainframe records. A text file can be downloaded to a PC in text mode with a carriage return (CR, hexadecimal value 0D) and line feed (LF, hexadecimal value 0A) marker appended to the end of each record as a record separator. This file can then be uploaded to a mainframe and reverted to its original form by detecting the CR/LF separator as a marker for the end of each record.

[0107] In general, any text file or hexadecimal file of fixed length records can be transferred to a PC in binary mode without appending a record end marker, and can be reverted to its original form when uploaded to a mainframe in the same binary mode. This is possible only because the length of all records are the same, and each record can be separated at the same length even without record separators. Text sequential datasets of undefined record format are rarely in use.

[0108] Mainframe load modules are not of fixed length records, they are of undefined record format and they

contain a lot of non-character hexadecimal data. As a result, mainframe load modules can not be reverted to their original form when they are uploaded back to a mainframe. Therefore, currently, mostly text datasets of fixed length records, e.g., usually members of text libraries, are delivered, member by member, to other mainframe data centers using PC-based data media.

[0109] The mainframe library is a partitioned dataset (PDS) and is equivalent to a PC directory or subdirectory which contains a large number of files as its components. The mainframe library also contains components so called members. There are only two types of mainframe libraries, a text library with only text members and a load library with only program load module members. Each member has records in it. Members of a library have common attributes, such as the record format, maximum record length, and length of each record. Each member can be handled as a separate sequential dataset. A single member of a text or load library is simple to process, but the processing of a library as a whole is not simple.

[0110] Text library members have fixed length records of 80-bytes long, and are used for storing program source codes, macros, procedures (PROCs), JCL statements, help texts, instructions, manuals, documents, and letters. Currently, only text library members are downloaded to a PC one member at a time, and copies of each text member are delivered using PC-based data media if necessary. A part or whole of a text library is still delivered using conventional magnetic tapes or cartridges exclusively.

[0111] The mainframe program load module is different from text files. The members of a load module library are all executable program load modules (the output of the IBM linkage editor IEWL, also called a load module, and the equivalent of a PC "program" file with .exe, .com, or .dll extensions). Each member has individual records in it. Each record of a load module contains mostly non-readable, non-character hexadecimal, interspersed with some readable characters. The length of any record can be different from that of any other record. The load library is defined only with the maximum record length (at least 256 bytes, but usually greater than 12,000 bytes, upto 32,760 bytes). Each record can be less than 256 bytes long or longer than 256 bytes, up to 32,760 bytes long.

[0112] These load modules must be transferred to a PC in binary mode due to the fact that it contains hexadecimal data. A single program load module is handled as a sequential dataset of undefined record, format, and it is not of fixed length records. Each record of different length must be separated from the next record by an end marker when downloaded to a PC, similar to the use of CR/LF as mentioned above in the case of the text mode transfer. However this end marker can be confused as normal hexadecimal data, and can not be handled correctly when uploaded to a mainframe. So, each record of a load module is concatenated at the end of the previous record without a separating marker when downloading. If this concatenated string of data, which does not possess a record separator, is uploaded to a mainframe in binary mode again, the end of each record can not be determined. Therefore the uploaded file will not have its original form, and will no longer be executable.

[0113] In summary, currently only a sequential dataset of fixed length records (e.g., a member of a text library), can be

delivered to other mainframe data centers using PC-related data media. A part or whole library can not be downloaded to a PC as a single unit, so magnetic tapes or cartridges are exclusively used for the distribution of a part or whole of a program load library or text library to other mainframe data centers.

SUMMARY OF INVENTION

[0114] A new mainframe program, PCFORM, converts the content of the whole or a part of any mainframe library of load modules or texts or any sequential dataset of fixed record format, undefined record format or variable record format to a sequential dataset of fixed length records (FIGS. 17-19). These converted data can be subsequently downloaded to a PC and reverted to their original form in the same or another mainframe data center, if uploaded from a PC.

[0115] This invention makes it possible for software text and load libraries and VSAM and DB2 datasets, which are sequential datasets of undefined record format, to be transferred to another data center. Any libraries or any sequential datasets can be transferred to another data center.

[0116] This invention allows the mainframe data transfer to be conducted using PC-based data media as an alternative to the conventional magnetic tape reels and tape cartridges. The new program PCFORM can be even distributed by a diskette, a recordable CD-ROM, over the world wide web, via FTP, or as an email attachment, and installed at the user's mainframe without having to use the aforementioned tapes. After the installation of PCFORM, any kind of library can be transported to that data center via any PC data media.

[0117] The PC-based data media mentioned above to be used in conjunction with this invention can include any of the following: 3.5-inch standard 1.44 MB floppy diskettes, 650-700 MB recordable CD-ROM's, 100 MB or 250 MB ZIP disks, 3.5-inch 120-Mbytes SuperDisk diskettes, email, Internet Website, TCP/IP FTP or TELNET, and any other PC data media technologies which will be widely used in the future, such as recordable DVD-ROM's.

[0118] This invention helps mainframe programmers download their mainframe texts and load modules from expensive mainframe disks to cheap and high-capacity PC hard disks. This allows mainframe programmers to use PC hard disk space to store backups of their mainframe work text data and load modules. This avoids the excessive buildup of backup libraries within mainframe disks, as well as the external accumulation of magnetic tapes.

[0119] By downloading the mainframe text library data to a PC, the mainframe programmers can perform routine text editing work on a stand-alone PC work station, instead of signing on to the mainframe. The copy of the downloaded PC file can be carried to a remote PC, where the mainframe programmer can perform any text editing.

[0120] The main benefits of this invention are derived from the facts that (1) PCs are ubiquitous, and familiar to most mainframe data center users, (2) most mainframe data centers already have a file transfer facility between their PC terminals and the mainframe, (3) a PC hard disk file of the downloaded mainframe dataset can be used repeatedly, (4) copying from a PC hard disk to PC-based data media is simpler, easier, and faster than creating a mainframe magnetic tape, (5) delivery is easy and simple, (6) emerging

CD-ROM, Internet, TCP/IP FTP, and email can be used, (7) PC data media can be a convenient means for the backup storage of a mainframe text library and load library, and (8) a PC, local or remote, can be used as a stand-alone working station for mainframe text editing, especially for program source codes.

[0121] The only requirement to be able to use this invention is that the sender and the receiver of this portable PC data media must have a file transfer facility between a mainframe and a PC. Then the program PCFORM can be installed on their mainframe system using the invention itself, without using any magnetic tapes from the start. After the program PCFORM is installed, a part or whole of any library or any sequential dataset can be downloaded to a PC, delivered to other mainframe computer data centers using PC-based data media, and uploaded from a PC in those receiving mainframe data centers.

[0122] There has been a need to better integrate emerging PC technology with mainframe technology, and to use the more convenient and cost effective PC-environment data transportation media in mainframe software libraries and data distribution. Also, there was a need to use cheap and high-capacity PC hard disks for storing mainframe text libraries, and use the PC as a stand-alone text editing work station for mainframe text library data in the PC hard disk.

[0123] The invention provides a method and apparatus for converting a part or whole of a mainframe software load library or text library or a mainframe sequential dataset of fixed record format, undefined record format or variable record format to a sequential dataset of fixed length records which can be distributed using PC-based data media to other mainframe computer data centers.

[0124] One embodiment of the invention provides a method and apparatus for reverting a mainframe software libraries and data delivered via PC-based data media to its original form when uploaded to a mainframe computer at the same site or at another site.

[0125] A further embodiment of the invention provides a method and apparatus for storing mainframe text libraries in a PC hard disk and working at a local or remote stand-alone PC for text editing of mainframe program source codes, macros, JCL streams, procs, manuals, documents, and letters which were downloaded from a mainframe.

BRIEF DESCRIPTION OF THE DRAWINGS

[0126] The invention and its particular features and advantages will become more apparent with examination of the following detailed description considered with the accompanying drawings, wherein:

[0127] FIG. 1 is a functional block diagram of the PC data media delivery of a single program load module when the sender has the program PCFORM, but the receiver does not;

[0128] FIG. 2 is a functional block diagram of the PC data media delivery of a program load library when both the sender and the receiver has the program PCFORM;

[0129] FIG. 3 is a functional block diagrams of the PC data media delivery of a text library when both the sender and the receiver has the program PCFORM;

[0130] FIG. 4 is a functional block diagrams of the PC data media delivery of a sequential dataset of fixed record

format, undefined record format or variable record format when both the sender and the receiver has the program PCFORM;

[0131] FIG. 5 shows how this invention can be used with the Internet world wide web download;

[0132] FIG. 6 shows how this invention can be used with the Internet email attachment file;

[0133] FIG. 7 shows how this invention can be used with the TCP/IP FTP protocol;

[0134] FIG. 8 is a functional block diagram of the current conventional magnetic tape delivery;

[0135] FIGS. 9-16 are the block diagrams of the program PCFORM;

[0136] FIG. 17 shows the conversion of an input record with a 4-bytes header and division into shorter records;

[0137] FIG. 18 shows the conversion of an input record with the division into shorter records and a 4-bytes header for each divided shorter record;

[0138] FIG. 19 shows the conversion of an input record with zero padding at the end of the input data; and

[0139] TABLE 1 is the listing of the source codes of the program PCFORM in assembler language.

DETAILED DESCRIPTION OF THE INVENTION

[0140] FIGS. 1 to 7 show four different situations within which this invention can be used with the PC-based data media. FIG. 8 shows the flow of events in the conventional magnetic tape delivery method. Before describing the block diagrams (FIGS. 1-7) of this invention, the flow diagram (FIG. 8) of the conventional magnetic tape delivery, is described first.

[0141] Conventional Delivery Using Magnetic Tapes

[0142] As an example explanation, let's assume that a software company, a sender 1, wants to deliver a whole text library, a program load module, and a whole load library 49 of a software package to a customer data center, a receiver 2, by tape 76. This explanation assumes that the name of a sender 1 is BSoft Co. The name of the software package to be delivered is assumed to be CPGM. Receiver 2 of this product CPGM will be referred to as C Co. The JCL streams of JCL Lists #1 and #2 explain how the events of FIG. 8 are actually handled in the mainframe data centers. The job control language (JCL) specifies the program name and the required files for the program. JCL streams or statements are entered to the operating system by the SUBMIT TSO command from the TSO/ISPF screen. The OS interprets these JCL statements, brings up the program from the step library or system load library, allocates the requested files, and passes the CPU control to the program.

[0143] A sender programmer 17 of a sending site 1 starts 71 a JCL stream of JCL List #1. An IBM utility program IEBCOPY 72 copies a whole text library, a program load module, and a whole load library 49 to tapes 76 as the output. A computer operator 69 mounts 73 the blank tape on the tape drive units when prompted by a system console 69. After IEBCOPY 72 finishes copying 74-75, system console 69 prompts the unloading 77 of tapes 76.

```

                                JCL List #1
                                Copy a text library, a load module
                                and a load library to magnetic tape

```

```

//jobname JOB (account),MSGCLASS=V,CLASS=L
//TEXTS EXEC PGM=IEBCOPY
//IN DD DISP=SHR,DSN=BSOFT.CPGM.TEXTLIB
//OUT DD DSN=TEXTLIB.TAPE,
//      DISP=(,KEEP,DELETE),
//      DCB=(BSOFT.CPGM.TEXTLIB),
//      UNIT=TAPE,
//      LABEL=(1,SL), dataset sequence no=1
//      VOL=(,RETAIN,,2) maximum 2 tapes
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//LOAD1 EXEC PGM=IEBCOPY
//IN DD DISP=SHR,DSN=BSOFT.CPGM.LOADLIB
//OUT DD DSN=PROGRAM1.TAPE,
//      DISP=(,KEEP,DELETE),
//      DCB=(BSOFT.CPGM.LOADLIB),
//      UNIT=TAPE,
//      LABEL=(2,SL), dataset sequence no=2
//      VOL=(,RETAIN,,,REF=*.TEXTS.OUT)
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
SELECT MEMBER=(PROGRAM1)
/*
//LOADS EXEC PGM=IEBCOPY
//IN DD DISP=SHR,DSN=BSOFT.CPGM.LOADLIB
//OUT DD DSN=LOADLIB.TAPE,
//      DISP=(,KEEP,DELETE),
//      DCB=(BSOFT.CPGM.LOADLIB),
//      UNIT=TAPE,
//      LABEL=(3,SL), dataset sequence no=3
//      VOL=(,RETAIN,,,REF=*.LOAD1.OUT)
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
//

```

[0144] Receiver programmer 28 receives delivered tapes 79. Receiver programmer 28 checks the volume serial number of tapes 79 and creates a JCL stream of JCL List #2. Receiver programmer at 28 brings received tapes 79 to a computer room 4 and starts 80 a loading JCL streams in JCL List #2 which requests a tape mount 82, and copies 83-84 the content of tapes 79 to destination libraries 58. After a load job 81 finishes, the computer operator unloads 85 tapes 79.

```

                                JCL List #2
                                Loading a text library, a load module,
                                and a load library from magnetic tape

```

```

//jobname JOB (account),MSGCLASS=V,CLASS=L
//TEXTS EXEC PGM=IEBCOPY
//IN DD DSN=TEXTLIB.TAPE,
//      DISP=OLD,
//      UNIT=TAPE,
//      LABEL=(1,SL), dataset sequence no=1
//      VOL=(,RETAIN,,,SER=123456)
/* destination text library

```

-continued

```

JCL List #2
Loading a text library, a load module,
and a load library from magnetic tape

```

```

//OUT DD DSN=C.CPGM.TEXTLIB,
// DISP=(,CATLG),
// DCB=(DSORG=PO,RECFM=FB,
// LRECL=80,BLKSIZE=8000),
// UNIT=SYSDA,
// SPACE=(CYL,(5,2,10)) depending on software
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//LOAD1 EXEC PGM=IEBCOPY
//IN DD DSN=PROGRAM1.TAPE,
// DISP=OLD,
// UNIT=TAPE,
// LABEL=(2,SL), dataset sequence no=2
// VOL=(,RETAIN,,,REF=*.TEXTS.IN)
/* destination load library
//OUT DD DSN=C.CPGM.LOADLIB,
// DISP=(,CATLG),
// DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760),
// UNIT=SYSDA,
// SPACE=(CYL,(20,5,50)) depending on software
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//LOADS EXEC PGM=IEBCOPY
//IN DD DSN=LOADLIB.TAPE,
// DISP=OLD,
// UNIT=TAPE,
// LABEL=(3,SL), dataset sequence no=3
// VOL=(REF=*.LOAD1.IN)
/* destination load library
//OUT DD DSN=C.CPGM.LOADLIB,
// DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(500,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
//

```

[0145] The tape delivery method is characterized by the operator interventions 73, 77, 82, 85, tapes 76, 79, two tape control units in two computer rooms 3, 4 and moving tapes between computer rooms 3, 4 and programmers' work areas 17, 28.

[0146] Delivery by the Present Invention

[0147] FIGS. 1-7 show four different situations within which this invention can be used with the PC-based data media. The floppy diskette is most common PC-based data media. The recordable CD-ROM is easy and simple to use and provides enough capacity for most mainframe data delivery situations. More and more mainframe data centers are in a situation in which they are ready to use the Internet and the TCP/IP FTP as the data transportation media. This invention does not require the involvement of any mainframe computer operators 69 and 70 nor magnetic tape drive units 76 and 79, as is required in the conventional tape delivery method shown in FIG. 8.

[0148] This new PC data media delivery method can be used in two different ways: when the receiver with and without the program PCFORM installed.

[0149] 1. When a Receiver Doesn't Have the Program PCFORM

[0150] FIG. 1 show the flow of the functional events in Procedure #5 when a sender 1 has a program PCFORM 13, but a receiver 2 doesn't have it. Sender 1 tries to send a copy of a single program load module 8 to another data center 2 which does not have the mainframe program PCFORM 13. In this case, receiver 1 can handle an individual program load module separately, but cannot handle a whole library of data.

Procedure #5

[0151] When the receiver doesn't have the program PCFORM

[0152] 1. A sender programmer at a PC 17 converts single mainframe program load module 8 to a download sequential dataset 16 of fixed record length of 256 bytes.

[0153] 2. Sender programmer at 17 downloads 18-23 the converted content in download sequential dataset 16 to PC 17 hard disk using Procedure #2, #3 or #4.

[0154] 3. Sender programmer at 17 chooses a PC data media 25, copies PC 17 hard disk file to a transportation media 25, and mails 26 it, or notifies receiver 2 that program load module 8 is in a sender's Website 62 or FTP site 65.

[0155] 4. A receiver programmer at PC 28 receives 26 a PC data media 27 and stores 29 it in PC 28 hard disk.

[0156] 5. Receiver programmer at 28 uploads 30-35 the received data from PC 28 to a mainframe upload sequential dataset 36 of the same record length, 256 bytes, as used for download sequential dataset 16 at sender 1.

[0157] 6. Receiver programmer at 28 converts the content of upload sequential dataset 36 to a member of a destination program load library 46 using an IBM utility program IEBCOPY 38 and an IBM linkage editor IEWL 43.

[0158] 2. When the Receiver has the Program PCFORM.

[0159] FIGS. 2-4 show the flow of functional events in Procedure #6 when both a sender 1 and a receiver 2 have program PCFORM 48, 55. Sender 1 can send a single load module, a part or whole of a load library, a part or whole of a text library or any sequential dataset 49 to receiver 2.

Procedure #6

[0160] When the Receiver has the Program PCFORM

[0161] 1. A sender programmer at a PC 17 converts 50-51 a whole or part of program load module library, a whole or part of text library or a sequential dataset 49 to a download sequential dataset 52 of fixed length record of 80-bytes long using program PCFORM 48.

[0162] 2. Sender programmer at 17 downloads 18-23 the content of download sequential dataset 52 to PC 17 hard disk using Procedure #2, #3 or #4.

[0163] 3. Sender programmer at 17 chooses a PC data transportation media 25, copies PC 17 hard disk file(s) to transportation media 25, and mails 26 them or notifies

receiver 2 that program load module(s) and/or text(s) are in a sender's Website 62 or FTP Server 65.

[0164] 4. A receiver programmer at a PC 28 receives 26 a PC data media 27 and stores 29 them in PC 28 hard disk.

[0165] 5. Receiver programmer at 28 uploads 30-35 the received data from PC 28 hard disk to a mainframe upload sequential dataset 53 of the same record length, 80 bytes, as used for download sequential dataset 52 of sender 1.

[0166] 6. Receiver programmer at 28 converts 56-57 the content of mainframe upload sequential dataset 53 to a part or whole of a destination program load module library, a part or whole of a destination text library or a destination sequential dataset of respective record format 58 accordingly using program PCFORM 55.

[0167] For both above two situations of Procedures #5 and #6, the first three steps are for sender 1 (i.e., a mainframe software vendor company) to create PC 17 hard disk file(s) from a whole or part of program load module library, a whole or part of text library or a sequential dataset 49 of a sender's mainframe system.

[0168] The last three steps are for receiver 2 (i.e., general user mainframe data centers) to convert data delivered in PC data media 27 to members of mainframe destination load module library or text library or a sequential dataset 58. At step 6, if program load module library and/or text library 58 already exists in a user data center 4, only the delivered members of libraries will be added or updated in place.

[0169] A downloaded mainframe data file in PC 17 hard disk can be copied to a Website 62 and downloaded by authorized mainframe users 28 via the Internet 60 as in FIG. 5. The program can be sent to each email user 28 directly as an email attachment as in FIG. 6. Or receiver programmer 28 can receive the content of sender's mainframe download dataset 16 via TCP/IP FTP as in FIG. 7.

[0170] The above two different situations, FIG. 1 and FIGS. 2-4, are described in more detail. All blocks, except blocks of TSO File Transfer and INDSFILE, in the figures represent different JCL streams, JCL Lists #3, #4, #6, #7, #9 and #12, which are explained in more detail. FIG. 1 is described first, and then FIGS. 2-4 are explained.

[0171] Detail Description of FIG. 1

[0172] FIG. 1 shows the functional block diagram when sender 1 has program PCFORM 13, but receiver 2 doesn't have the mainframe program PCFORM. In this situation, receiver 2 can only handle a single load module. This diagram explains how to send single program load module 8 to a mainframe user data center 2 via floppy diskettes or recordable CD-ROM 25, 27 without the involvement of magnetic tapes, and how to install the program into a mainframe destination load library 46. This diagram also shows how the new program PCFORM can be delivered to other software companies and general mainframe user companies without the need for magnetic tapes at the beginning.

[0173] For this example explanation, the JCL streams assume that the name of sender 1 is ASoft Co., the developer of the program PCFORM. The name of the program to be delivered is assumed to be PCFORM. In the example, receiver does not have the program PCFORM. Receiver 2 of this product PCFORM will be referred to as C Co., with a

mainframe data center 4. Program PCFORM 8 is assumed to be in the program library ASOFT.LOADLIB at sender 1. The delivered PCFORM will be installed in C Co.'s destination load library, C.ASOFT.LOADLIB 46.

[0174] Download Procedure of the Sender in FIG. 1

[0175] The left side of FIG. 1 describes the download procedure for sender 1 to create portable floppy diskette or recordable CD-ROM 25.

[0176] The first step of creating download sequential dataset 16 is done by use of an IBM linkage editor program IEWL 7. The maximum record length of input load module PCFORM 8 in the COPYLIB load library is usually very large, e.g., up to 32,760 bytes. The maximum record length of an output temporary load module member 11 in a SYSLMOD temporary load library is defined as 256 bytes as a default: this is the recommended length. The BLKSIZE value of SYSLMOD temporary load library 11 must be 256 bytes or greater. But the BLKSIZE of 256 bytes is recommended. Step 7 converts 9-10 original load module PCFORM 8 into new temporary load module 11 of shorter records. IBM linkage editor IEWL 7 performs this conversion when sender programmer 17 enters 6 a JCL stream of JCL List #3 to the operating system.

```
JCL List #3  
Creating a temporary load module of shorter length
```

```
//jobname JOB (account),MSGCLASS=V,CLASS=L
//LKED EXEC PGM=IEWL,PARM='MAP,XREF,LET,LIST'
//SYSPRINT DD SYSOUT=*
//SYSLOUT DD SYSOUT=*
//SYSUT1 DD DSN=&&SYSUT1,
// UNIT=SYSDA,SPACE=(CYL,(2,1))
//COPYLIB DD DISP=SHR,DSN=ASOFT.LOADLIB
//SYSLMOD DD DSN=TEMP.LOADLIB,
// DISP=(,CATLG),
// DCB=(DSORG=PO,RECFM=U,
// LRECL=0,BLKSIZE=256)
// UNIT=SYSDA,SPACE=(CYL,(1,1,3))
//SYSLIN DD *
// ENTRY PCFORM
// INCLUDE COPYLIB (PCFORM)
// NAME TEMPTEMP
//
```

[0177] Here, the SYSLIN control information may be adjusted for each program load module 8 and receiver 2 of this program will use this SYSLIN control information when he uploads the program delivered. Depending on the software program, the first SYSLIN line ENTRY control statement may or may not be required, and can have a different name from that of the program.

[0178] Output temporary load module TEMPTEMP 11 in the SYSLMOD library is still an executable load module, which means that the length of each record in the module may be different from the length of others.

[0179] At the next step, program PCFORM 13 converts 14-15 temporary load module TEMPTEMP 11 to download sequential dataset 16 of uniform fixed length records. This conversion is necessary because only uniform fixed length records of a load module can be reverted to the original load module status when they are downloaded to a PC and uploaded to a mainframe back again. Program PCFORM 13

is executed by entering **12** a JCL stream of JCL List #4 to the OS/390 by sender programmer at **17**:

```

                JCL List #4
                Creating the revertible sequential
                dataset from the temporary load module


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//CONVERT EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=ASOFT.LOADLIB
//IN DD DISP=SHR,DSN=TEMP.LOADLIB(TEMPTEMP)
//OUT DD DSN=DOWNLOAD.WORK,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,
//          LRECL=256,BLKSIZE=2560),
//      UNIT=SYSDA,SPACE=(CYL,(1,1))
//CONTROL DD *
DUMP
//

```

[0180] The LRECL value of output download sequential dataset **16** specified by OUT ddname must be the same as the BLKSIZE value (256 bytes is recommended and is used as the default) of IN temporary load library **11** created by a JCL stream of JCL List #3. The disk SPACE value of OUT download sequential dataset **16** is approximately the same as the size of original program load module **8**, when the value of its LRECL is near **256**.

[0181] Original program load module library **8** itself can be used directly as the IN dataset of above JCL stream (JCL List #4). In this case, linkage editor IEWL **7** step which was necessary in the previous case can be eliminated. But because the LRECL value of OUT download sequential dataset **16** must be the same as the BLKSIZE value of the IN dataset, and due to the large BLKSIZE of original load library **8** (e.g., up to 32,760), the LRECL value of OUT download sequential dataset **16** becomes correspondingly very large. Thus the total output size to transfer between a mainframe and a PC, and between two mainframe data centers can be very large. This is not recommended. The smaller the LRECL value of the OUT dataset, the smaller the amount of the data which needs to be transferred. This is why the minimum allowable value, 256 bytes, is recommended as the default value.

[0182] Now, download sequential dataset **16** is ready to be transferred to a PC.

[0183] Sender programmer at **17** transfers **18-23** the content of mainframe download sequential dataset, DOWNLOAD.WORK, **16** to PC **17** as C:\pcform.exe in Binary mode using Procedure #2, #3 or #4.

[0184] Sender programmer at **17** copies **24** C:\pcform.exe, to a PC data transportation medium **25**, e.g., a floppy diskette, A:\pcform.exe, or a recordable CD-ROM, D:\pcform.exe.

[0185] This dataset data can include letters, installation guides, installation JCLs, help information, manuals, and other documents.

[0186] Sender **1** mails **26** diskette or CD-ROM **25** to receiver data center **2**.

[0187] A downloaded load module file in PC **17** hard disk can be copied to a Website **62** and downloaded by authorized

mainframe users **28** via the Internet **60** as in **FIG. 5**. The program can be sent to each email user **28** directly as an email attachment as in **FIG. 6**. Or receiver programmer **28** can receive the content of sender's mainframe download dataset **16** via TCP/IP FTP as in **FIG. 7**.

[0188] Upload Procedure of the Receiver in **FIG. 1**

[0189] The right side of **FIG. 1** describes the functional event flow at receiver's mainframe site **2**, referred to as the upload procedure for the user data center.

[0190] Receiver programmer at **28** of mainframe site **2** receives floppy diskettes or CD-ROM **27** mailed **26** from sender **1** and saves **29** the content of diskettes or CD-ROM **27** to the hard disk of PC **28**. Receiver programmer at **28** copies A:\pcform.exe or D:\pcform.exe to C:\ASOFT\pcform.exe.

[0191] Receiver programmer at **28** submits a JCL stream in JCL List #5 to the mainframe operating system to create upload sequential dataset **36**, destination program load library **46**, and destination text library **46** which will receive the data delivered.

[0192] The LRECL value 256 of FILE1 upload sequential dataset **36** must be the same value as the LRECL value (256 bytes recommended and used as the default) of download sequential dataset **16** of sender **1**.

[0193] The BLKSIZE value 32,760 of FILE2 destination load library **46** is the maximum allowed value, and is the value which is recommended to use for the transfer. This value can be changed to any number equal to 256 bytes or greater.

```

                JCL List #5
                Creating the receiving datasets


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//NEWDS EXEC PGM=IEFBR14
//FILE1 DD DSN=UPLOAD.WORK,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,
//          LRECL=256,BLKSIZE=2560),
//      UNIT=SYSDA,SPACE=(CYL,(1,1))
//* destination load library
//FILE2 DD DSN=C.ASOFT.LOADLIB,
//      DISP=(,CATLG),
//      DCB=(DSORG=PO,RECFM=U,
//          BLKSIZE=32760),
//      UNIT=SYSDA,
//      SPACE=(CYL,(1,1,5)) depending on software
//* destination text library
//FILE3 DD DSN=C.ASOFT.TEXTLIB,
//      DISP=(,CATLG),
//      DCB=(DSORG=PO,RECFM=FB,
//          LRECL=80,BLKSIZE=8000),
//      UNIT=SYSDA,
//      SPACE=(CYL,(5,2,10)) depending on software
//

```

[0194] Receiver programmer at **28** transfers **30-35** delivered data in PC **28**, C:\ASOFT\pcform.exe, to mainframe upload sequential dataset, UPLOAD.WORK, **36** in Binary mode using Procedures #2, #3 or #4.

[0195] Receiver programmer at **28** submits **37** a JCL stream in JCL List #6. JCL List #6 then calls IBM dataset utility program IEBGENER **38** and copies **39-40** the content

of SYSUT1 upload sequential dataset 36 into a temporary member TEMPTMP 41 of SYSUT2 destination load library 46.

```

JCL List #6
Creating a temporary member in a load library


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DISP=SHR,DSN=UPLOAD.WORK
//SYSUT2 DD DISP=SHR,DSN=C.ASOFT.LOADLIB(TEMPTMP)
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//

```

[0196] Temporary output member TEMPTMP 41 of SYSUT2 destination load library is not executable. Its only purpose is to be used as the input for linkage editor IEWL 43 in a next JCL stream of JCL List #7.

[0197] Receiver programmer at 28 then submits 42 a JCL stream in JCL List #7 which calls IBM utility program linkage editor IEWL and converts 44-45 non-executable member TEMPTMP 41 of COPYLIB library to an executable load module member PCFORM 46 in the SYSLMOD destination load library.

```

JCL List #7
Creating the executable load module
member from the temporary member


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//LKED EXEC PGM=IEWL,PARM='MAP,XREF,LET,LIST'
//SYSPRINT DD SYSOUT=*
//SYSLOUT DD SYSOUT=*
//SYSUT1 DD DSN=&&SYSUT1,UNIT=SYSDA,
//          SPACE=(CYL,(1,1))
//COPYLIB DD DISP=SHR,DSN=C.ASOFT.LOADLIB
//SYSLMOD DD DISP=SHR,DSN=C.ASOFT.LOADLIB
//SYSLIN DD *
          ENTRY PCFORM
          INCLUDE COPYLIB(TEMPTMP)
          NAME PCFORM
//

```

[0198] JCL List #7 creates an executable final load module member PCFORM 46, which was delivered in a converted form via diskette or CD-ROM 27, in SYSLMOD destination load library 46.

[0199] This example description illustrates how single load module PCFORM 46 was delivered via floppy diskette or CD-ROM 27, without having to use conventional magnetic tape 79, and is installed just using the available IBM utility programs at mainframe user data center 2, even without the help of the program PCFORM.

[0200] In summary, this invention will now allow a general mainframe user company, e.g., C Co., to receive any library and any sequential datasets from any mainframe software company, using any kind of PC-based data media, without any need whatsoever for the conventional magnetic tapes which are being currently used for such transfers.

[0201] Detail Description of FIGS. 2 and 3

[0202] A situation in which both sender 1 and receiver 2 have program PCFORM 48, 55 will now be explained. In

this scenario both sender 1 (a software company) and receiver 2 (a general mainframe data center) have installed program PCFORM 48, 55 by the method depicted in FIG. 1, and as explained above.

[0203] FIGS. 2 and 3 diagrams explain how to send and install a whole or part of a load library or a text library of a mainframe software company 1 to a mainframe user data center 2 via floppy diskettes or recordable CD-ROM 25 without the use of magnetic tapes. FIG. 4 diagram show how a mainframe data center 1 send a sequential dataset of fixed record format, undefined record format or variable record format, e.g., copy datasets of VSAM datasets or DB2 datasets, to another mainframe data center using PC-based data media.

[0204] FIGS. 2 and 3 are explained first and FIG. 4 will be explained later.

[0205] The JCL streams of this diagram also assumes for the sake of convenience, as in the FIG. 1 explanation, that the name of sender 1, a mainframe software company, is assumed to be BSoft Co. A client company 2 of BSoft Co. is assumed to be C Co. The name of a mainframe software package 49 to be delivered to C Co. is CPGM. The load modules and related texts of the mainframe software package, CPGM, reside in the libraries BSOFT.CPGM.LOADLIB and BSOFT.CPGM.TEXTLIB at BSoft Co. They will be installed into the destination libraries C.CPGM.LOADLIB and C.CPGM.TEXTLIB of C Co. Program PCFORM is assumed in the program library BSOFT.ASOFT.LOADLIB at BSoft Co. and C.ASOFT.LOADLIB at C Co.

[0206] This example diagram is based on the use of 3.5-inch standard 1.44 MB floppy diskettes 25 and 27. Using recordable CD-ROM media is simpler than floppy diskettes.

[0207] Download Procedure of the Sender in FIGS. 2 and 3

[0208] We will begin by detailing the download procedure for sender 1 to create portable floppy diskettes 25 containing the data to be transferred.

[0209] Sender programmer at 17 creates a JCL stream in JCL List #8 for creating download sequential dataset 52 of fixed length records.

```

JCL List #8
Creating the download sequential dataset


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//NEWDS EXEC PGM=IEFBRI4
//FILE1 DD DSN=DOWNLOAD.WORK,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,
//          LRECL=80,BLKSIZE=8000),
//          UNIT=SYSDA,SPACE=(CYL,(10,10))
//

```

[0210] The primary and secondary quantity of CYLinders may be different depending on the size of each target load library or text library to be delivered, and whether a whole or part of the library or just a single member is to be delivered. One cylinder can hold approximately 650 K bytes of data. In the above example the SPACE parameter speci-

fies a maximum of 160 cylinders and an approximate maximum of 104 million bytes (equivalent to approximately 7 2400-ft magnetic tape reels). Download sequential dataset 52 is used repeatedly, so SPACE parameter should be allocated a high enough value to fit all the libraries to be sent out.

[0211] The LRECL value of texts is fixed at 80 bytes. There is no variety and thus no confusion about the LRECL value for a text library. But for program load modules, sender programmer at 17 can choose any fixed LRECL value of 80 or greater for download sequential dataset, DOWNLOAD.WORK, 52. Receiver programmer at 28 of main-frame site 2 must use this same LRECL value to create upload sequential dataset 53. In order to avoid any confusion, the LRECL value of download sequential dataset 52 can be fixed at 80-bytes long as a default value for both the program load modules and texts. The LRECL value 80 is strongly recommended. And this default value of 80 bytes length is used in this example explanation. This dataset of 80 bytes record length can be used for both text libraries and load libraries.

[0212] When program load modules are sent out to users, sender programmer 17 may notify receiver programmer 28 about the BLKSIZE value of original load module library, e.g., BSOFT.CPGM.LOADLIB 49. Receiver programmer 28 can create destination program load module library 58 with the BLKSIZE value suggested by sender programmer 17, or greater up to the maximum allowed BLKSIZE value of 32,760 which can receive any BLKSIZE value of original load library 49.

[0213] Sender programmer at 17 submits the above IBM dataset utility program IEFBR14 JCL stream (JCL List #8) to create download sequential dataset, DOWNLOAD.WORK 52.

[0214] Download sequential dataset 52 and other download work datasets which will be created later will be used repeatedly for each library download. Therefore the full procedure, from the submission 47 of PCFORM program 48 to the completion 23 of the file transfer to a PC, must be done separately for each library whether it is a load library or a text library. A library is downloaded to a PC one library at a time.

[0215] Sender programmer at 17 creates a JCL stream (JCL List #9) which will bring up program PCFORM 48 from the STEPLIB library. The IN DD statement identifies original text or load library 49 to be delivered. The OUT dataset is download sequential dataset 52.

```

JCL List #9
Dumping the content of a load library to the download sequential dataset
//jobname JOB (account),MSGCLASS=V,CLASS=L
//PCFORM EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=BSOFT.ASOFT.LOADLIB
//SYSPRINT DD SYSOUT=*
//* original library
//IN DD DISP=SHR,DSN=BSOFT.CPGM.LOADLIB
//* download dataset
//OUT DD DISP=SHR,DSN=DOWNLOAD.WORK

```

-continued

```

JCL List #9
Dumping the content of a load library to the download sequential dataset
//CONTROL DD *
DUMP
//

```

[0216] The DUMP CONTROL command without any member names makes program PCFORM 48 dump 50-51 whole records in IN input original text or load library 49 to OUT output download sequential dataset 52.

[0217] The CONTROL control command DUMP converts the content of IN original library 49 to OUT download sequential dataset 52 records. The DUMP command starts at column 1 and can be followed by any number of library member names; in this case only the named members will be dumped to download sequential dataset 52. More than one member list lines can follow the DUMP control command line, but the first column must be blank. Using DUMP control command without any member names, as shown in the above sample JCL stream, will convert the whole content of original library 49 to download sequential dataset 52 records.

```

JCL Note #1
Examples of free format DUMP command syntax
(1). DUMP
(2). DUMP PROGRAM1 Program2
        program3,PROGRAM4,
        PROGRAM5 program6,
        PROGRAM7,PROGRAM8
(3). DUMP
        PROGRAM1 PROGRAM2
        PROGRAM3, PROGRAM4
        PROGRAM5
        PROGRAM6,PROGRAM7
(4). DUMP (
        JCL1, JCL2, JCL3, HELP1,
        HELP2, UPLD1,
        Manual1
        Manual2
        ERRORMSG ,VENDOR
)

```

[0218] Let's start with the download of load library 49 first.

[0219] The download procedure of a text library will be repeated after the completion of the load library download.

[0220] Sender programmer at 17 submits 47 above PCFORM program JCL stream (JCL List #9) to dump the software package in original load library 49 to download sequential dataset 52.

[0221] If output download sequential dataset 52 is too big for a standard 1.44 MB floppy diskette, the records in download sequential dataset 52 must be divided into smaller sequential datasets in order to use 1.44 MB floppy diskettes as the transportation media. This division is necessary only for 1.44 MB floppy diskettes. If a recordable CD-ROM is used, this division is not necessary. Also, if the data is to be transferred over the world wide web or the TCP/IP FTP, this division is not necessary.

[0222] If a text member or a part or whole of a text library is downloaded to a PC, sometimes it can be edited in the PC using a PC word processor or an ASCII editor. Therefore it is handy to keep the text file small enough to edit in the PC using ASCII editor. Some ASCII editors can not handle larger files.

[0223] For load modules, this division can be done only in the mainframe. But for texts, this division can be done in the mainframe or in a PC with PC word processors or ASCII text editors.

[0224] One 3.5-inch floppy diskette can deliver 1.4 million bytes or up to approximately 17,000 records of 80-byte length mainframe records. But let's use the number 15,000 as an example for simplicity. If the library to be downloaded is a text library, it is recommended to divide the library into smaller datasets which are easier for PC text editors to handle.

[0225] If download sequential dataset, DOWNLOAD.WORK, 52 contains more than 15,000 records of 80-byte long, it is necessary to create more than one smaller temporary work sequential datasets, e.g., DOWNLOAD.WORK1 and DOWNLOAD.WORK2, and so on, until the whole library can be divided into approximately 15,000 80-byte records per each smaller dataset. These temporary smaller datasets are created using the same DCB parameters as download sequential dataset 52, that is, DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=8000, but with a different SPACE parameter, SPACE=(CYL,(1,1)) for example.

[0226] And sender programmer at 17 divides the content of download sequential dataset, DOWNLOAD.WORK, 52 into smaller datasets of 15,000 records for each dataset using the TSO/ISPF EDIT panel or another software program. For an explanation, let's assume that download sequential dataset 52 was divided into four smaller datasets, DOWNLOAD.WORK1, DOWNLOAD.WORK2, DOWNLOAD.WORK3, and DOWNLOAD.WORK4.

[0227] Now, download sequential dataset(s) 52 is ready to be downloaded to a PC.

[0228] Sender programmer at 17 transfers 18-23 mainframe download sequential datasets 52 to the hard disk of PC 17 individually in Binary mode using Procedure #2, #3 or #4:

[0229] 1. if download sequential dataset 52 itself is to be downloaded 18-23, download DOWNLOAD.WORK to the PC file C:\CPGM.exe,

[0230] or

[0231] 2. if download sequential dataset was divided into smaller datasets,

[0232] download DOWNLOAD.WORK1 to C:\CPGM1.exe,

[0233] download DOWNLOAD.WORK2 to C:\CPGM2.exe,

[0234] download DOWNLOAD.WORK3 to C:\CPGM3.exe,

[0235] download DOWNLOAD.WORK4 to C:\CPGM4.exe.

[0236] Now, whole original load library 49 is stored in the hard disk of PC 17. It is ready to be delivered to any client user data center 2, e.g., including C Co. in this example.

[0237] Now, let's repeat the full download procedure for text library 49. As mentioned before, the download for each individual library starts with the submission 47 of program PCFORM JCL stream (JCL List #10).

[0238] This time, the IN dataset of PCFORM program 48 is a text library 49. Same download sequential dataset 52 is used repeatedly as the OUT dataset. So the previous content of download sequential dataset 52 will be overwritten with new content every time after this step 48.

```
JCL List #10
Dumping the content of a text library
to the download sequential dataset
```

```
//jobname JOB (account),MSGCLASS=V,CLASS=L
//PCFORM EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=BSOFT.ASOFT.LOADLIB
//SYSPRINT DD SYSOUT=*
/* original library
//IN DD DISP=SHR,DSN=BSOFT.CPGM.TEXTLIB
/* download dataset
//OUT DD DISP=SHR,DSN=DOWNLOAD.WORK
//CONTROL DD *
DUMP
//
```

[0239] After PCFORM program 48 is finished successfully, the content of download sequential dataset 52 can be divided into smaller work datasets, e.g., DOWNLOAD.WORK1, DOWNLOAD.WORK2, DOWNLOAD.WORK3, DOWNLOAD.WORK4, and so on, using the same method as used for the load library above.

[0240] As the final step of the text library download, sender programmer 17 transfers 18-23 the mainframe datasets to the hard disk of PC 17 individually in Text mode using Procedure #2, #3 or #4:

[0241] 1. if download sequential dataset 52 itself is to be downloaded, download DOWNLOAD.WORK to the PC file C:\CPGM.txt,

[0242] or

[0243] 2. if download sequential dataset 52 was divided into smaller work datasets,

[0244] download DOWNLOAD.WORK1 to C:\CPGM1.txt,

[0245] download DOWNLOAD.WORK2 to C:\CPGM2.txt,

[0246] download DOWNLOAD.WORK3 to C:\CPGM3.txt,

[0247] download DOWNLOAD.WORK4 to C:\CPGM4.txt.

[0248] Now, whole program load library and/or whole text library 49 are stored in the hard disk of PC 17. They are ready to be delivered to any client user data center 2, e.g., C Co. in this example.

[0249] Sender programmer at 17 copies 24 all parts of the downloaded libraries from the hard disk of PC 17 to PC data media 25. In the case of the example below, the libraries are copied to 1.44 MB floppy diskettes:

- [0250] copy C:\CPGM1.exe to A:\CPGM1.exe
- [0251] copy C:\CPGM2.exe to A:\CPGM2.exe
- [0252] copy C:\CPGM3.exe to A:\CPGM3.exe
- [0253] copy C:\CPGM4.exe to A:\CPGM4.exe
- [0254] copy C:\CPGM1.txt to A:\CPGM1.txt
- [0255] copy C:\CPGM2.txt to A:\CPGM2.txt
- [0256] copy C:\CPGM3.txt to A:\CPGM3.txt
- [0257] copy C:\CPGM4.txt to A:\CPGM4.txt

[0258] This data will be accompanied with letters, installation guides, installation JCLs, help information, manuals, and other documents.

[0259] Now, sender programmer 17 mails diskettes 25 which were created.

[0260] As like the explanation of FIG. 1 diagram, downloaded load module and text files in PC 17 hard disk can be copied to a Website 62 and downloaded by authorized mainframe users 28 via the Internet 60 as in FIG. 5. The files can be sent to each email user 28 directly as an email attachments as in FIG. 6. Or !5 receiver programmer 28 can receive the content of sender's mainframe download dataset 16 via TCP/IP FTP as in FIG. 7.

[0261] Upload Procedure of the Receiver in FIGS. 2 and 3

[0262] The right side of the FIGS. 2 and 3 show the upload procedure for each user data center which received the diskettes of a library.

[0263] Receiver programmer 28 receives 26 floppy diskettes 27 and saves 29 them onto PC 28 hard disk as:

- [0264] copy A:\CPGM1.exe to C:\BSOFT\CPGM1.exe
- [0265] copy A:\CPGM2.exe to C:\BSOFT\CPGM2.exe
- [0266] copy A:\CPGM3.exe to C:\BSOFT\CPGM3.exe
- [0267] copy A:\CPGM4.exe to C:\BSOFT\CPGM4.exe
- [0268] copy A:\CPGM1.txt to C:\BSOFT\CPGM1.txt
- [0269] copy A:\CPGM2.txt to C:\BSOFT\CPGM2.txt
- [0270] copy A:\CPGM3.txt to C:\BSOFT\CPGM3.txt
- [0271] copy A:\CPGM4.txt to C:\BSOFT\CPGM4.txt.

[0272] Receiver programmer 28 submits a JCL stream in JCL List #11 to create required datasets: destination load library 58, destination text library 58, and upload sequential datasets 53, e.g., UPLOAD.WORK1, UPLOAD.WORK2, UPLOAD.WORK3, and UPLOAD.WORK4, to receive the data delivered.

```
JCL List #11
Creating the receiving datasets
```

```

//jobname JOB (account),MSGCLASS=V,CLASS=L
//NEWDS EXEC PGM=IEFBR14
/* destination text library
//DD1 DD DSN=C.CPGM.TEXTLIB,
// DISP=(,CATLG),
// DCB=(DSORG=PO,RECFM=FB,
// LRECL=80,BLKSIZE=8000),
// UNIT=SYSDA,
// SPACE=(CYL,(10,10,20)) depending on software
/* destination program library
//DD2 DD DSN=C.CPGM.LOADLIB,
// DISP=(,CATLG),
// DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760),
// UNIT=SYSDA,
// SPACE=(CYL,(10,10,20)) depending on software
//DD3 DD DSN=UPLOAD.WORK1,
// DISP=(,CATLG),
// DCB=(DSORG=PS,RECFM=FB,
// LRECL=80,BLKSIZE=800),
// UNIT=SYSDA,SPACE=(CYL,(1,1))
/* for Cpgm2.exe or .txt
//DD4 DD DSN=UPLOAD.WORK2,
// DISP=(,CATLG),
// DCB=(DSORG=PS,RECFM=FB,
// LRECL=80,BLKSIZE=800),
// UNIT=SYSDA,SPACE=(CYL,(1,1))
/* for Cpgm3.exe or .txt
//DD5 DD DSN=UPLOAD.WORK3,
// DISP=(,CATLG),
// DCB=(DSORG=PS,RECFM=FB,
// LRECL=80,BLKSIZE=800),
// UNIT=SYSDA,SPACE=(CYL,(1,1))
/* for Cpgm4.exe or .txt
//DD6 DD DSN=UPLOAD.WORK4,
// DISP=(,CATLG),
// DCB=(DSORG=PS,RECFM=FB,
// LRECL=80,BLKSIZE=800),
// UNIT=SYSDA,SPACE=(CYL,(1,1))
//

```

[0273] The LRECL value of upload sequential datasets, UPLOAD.WORKn 53 must be the same value as LRECL value of download sequential dataset, DOWNLOAD.WORK, 52 at sender's site 1. The LRECL value of 80 bytes is the default value for the sender. But sender 1 may use a different value, so receiver 2 must be careful about this LRECL value.

[0274] The disk space parameters of destination load library and destination text library 58 must be carefully decided. This space must be sufficient to receive all the data delivered. Mostly sender 1 supplies this information. Upload sequential datasets 53 are used repeatedly for each upload of both the text and load library data, so it is recommended to allocate enough space to fit all situations.

[0275] At the above, the BLKSIZE value 32,760 of destination load library, C.CPGM.LOADLIB, 58 is the maximum allowed value, and is recommended and set as the default value. This value can be changed to any value equal to or greater than the BLKSIZE value of original load library 49 at sender 1, which is usually supplied by sender 1.

[0276] Like the download procedure of each library, the whole upload procedure, from the first step 30 of the file transfer step 30-35 to the completion of loading 55 the final

library, is also performed repeatedly for each individual library delivered. Each library must be uploaded separately individually.

[0277] Let's start with the upload of the load library data delivered. The upload procedure for the text library data will be repeated later after the completion of the upload of the load library data.

[0278] Receiver programmer 28 now transfers 30-35 PC 28 files of original load library 49 to mainframe upload sequential datasets 53 in binary mode using Procedure #2, #3 or #4.

- [0279] transfer C:\BISOFT\CPGM1.exe to
UPLOAD.WORK1.
- [0280] transfer C:\BISOFT\CPGM2.exe to
UPLOAD.WORK2.
- [0281] transfer C:\BISOFT\CPGM3.exe to
UPLOAD.WORK3.
- [0282] transfer C:\BISOFT\CPGM4.exe to
UPLOAD.WORK4.

[0283] Now, all the copies of original load library 49 are transferred to a receiver's mainframe 4.

[0284] Receiver programmer 28 has to load 56-57 all the contents of upload sequential dataset 53 to destination load library 58 by running 54 program PCFORM 55. APCFORM program JCL stream in JCL List #12 is submitted 54 and program PCFORM 55 loads 56-57 all the data in IN upload sequential datasets 53 to OUT destination program load library 58.

JCL List #12
Loading the delivered load library data to a load library

```

//jobname JOB (account),MSGCLASS=V,CLASS=L
//PCFORM EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=C.ASOFT.LOADLIB
//SYSPRINT DD SYSOUT=*
//IN DD DISP=SHR,DSN=UPLOAD.WORK1
// DD DISP=SHR,DSN=UPLOAD.WORK2
// DD DISP=SHR,DSN=UPLOAD.WORK3
// DD DISP=SHR,DSN=UPLOAD.WORK4
/* destination library
//OUT DD DISP=SHR,DSN=C.CPGM.LOADLIB
//CONTROL DD *
LOAD
//

```

[0285] Here, the CONTROL control command LOAD converts the content of upload sequential datasets 53 to the members of destination library 58. The LOAD command starts at column 1 and can be followed by any number of library member names. More than one member list lines can follow the LOAD control command line, but the first column of the following lines must be blank. In this case, only the named members are picked up from upload sequential datasets 53 and loaded to destination library 58. The LOAD control command without any member names, as shown as in the above sample JCL stream, will load the whole content of upload sequential datasets 53 to destination library 58.

JCL Note #2
Example of free format LOAD command syntax

```

(1). LOAD
(2). LOAD PROGRAM1 PROGRAM2
PROGRAM3,Program4,
PROGRAM5 program6,
PROGRAM7,PROGRAM8
(3). LOAD
PROGRAM1 PROGRAM2
program3, PROGRAM4
PROGRAM5
PROGRAM6,PROGRAM7
(4). LOAD (
text jcl2
source3
Macro4
Source5 ,SOURCE6 )

```

[0286] Now, delivered original load library or load modules 49 were loaded 56-57 into destination program load module library 58. This was done without the use of magnetic tape.

[0287] The upload of a part or whole of a text library also can be done in just the same way. At first, the PC files of delivered text data are transferred 30-35 to mainframe upload sequential datasets 53 and the content of upload sequential datasets 53 are loaded 56-57 to destination text library 58 by a program PCFORM JCL stream (JCL List #13).

[0288] Receiver programmer 28 transfers the PC files of original text library 49 to mainframe upload sequential datasets 53 in Text mode using Procedure #2, #3 or #4.

- [0289] transfer C:\BISOFT\CPGM1.txt to UPLOAD-
.WORK1.
- [0290] transfer C:\BISOFT\CPGM2.txt to UPLOAD-
.WORK2.
- [0291] transfer C:\BISOFT\CPGM3.txt to UPLOAD-
.WORK3.
- [0292] transfer C:\BISOFT\CPGM4.txt to UPLOAD-
.WORK4.

[0293] Receiver programmer 28 loads 56-57 all the transferred content to destination text library 58 by running 54 program PCFORM JCL stream shown in JCL List #13 which is just the same as above load library upload procedure, except for the OUT library name. The IN dataset is upload sequential dataset 53 and OUT dataset is final destination text library 58.

JCL List #13
Loading the delivered text library data to a text library

```

//jobname JOB (account),MSGCLASS=V,CLASS=L
//PCFORM EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=C.ASOFT.LOADLIB
//SYSPRINT DD SYSOUT=*
//IN DD DISP=SHR,DSN=UPLOAD.WORK1
// DD DISP=SHR,DSN=UPLOAD.WORK2
// DD DISP=SHR,DSN=UPLOAD.WORK3
// DD DISP=SHR,DSN=UPLOAD.WORK4
/* destination text library
//OUT DD DISP=SHR,DSN=C.CPGM.TEXTLIB

```

-continued

```

                JCL List #13
    Loading the delivered text library data to a text library


---


//CONTROL DD *
LOAD
//

```

[0294] Text members or a whole text library were loaded 56-57 into destination text library 58. JCL Lists #12 and #13 are similar except the OUT datasets.

[0295] Delivery of Various Sequential Datasets in FIG. 4

[0296] Any sequential dataset of fixed record format, undefined record format or variable record format can be transferred to another mainframe using PC-based data media and can be reverted to its original form when uploaded, as depicted in FIG. 4.

[0297] The events in FIG. 4 can be explained using JCL streams of JCL Lists #14 and #15, which are similar to JCL Lists #9 or #10 and JCL Lists #12 or #13 with only small differences. The IN dataset in JCL stream of JCL List #14 at the sender's site and the OUT dataset in JCL List #15 at the receiver's site are sequential datasets of fixed record format, undefined record format or variable record format respectively. The block size of both IN and OUT sequential datasets must be the same when they are of undefined record format or variable record format. The logical record length of both IN and OUT sequential datasets must be same when they are of fixed record format. Download and upload procedures of FIG. 4 are exactly the same as in FIGS. 2 and 3.

```

                JCL List #14
    Dumping the content of a sequential
    dataset to the download sequential dataset


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//PCFORM EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=BSOFT.ASOFT.LOADLIB
//SYSPRINT DD SYSOUT=*
//IN DD DISP=SHR,DSN=a_F/U/V_dataset_name
//OUT DD DISP=SHR,DSN=DOWNLOAD.WORK
//CONTROL DD *
DUMP
//

```

```

                JCL List #15
    Loading the delivered data to a destination sequential
    dataset


---


//jobname JOB (account),MSGCLASS=V,CLASS=L
//PCFORM EXEC PGM=PCFORM
//STEPLIB DD DISP=SHR,DSN=C.ASOFT.ASOFT.LOADLIB
//SYSPRINT DD SYSOUT=*
//IN DD DISP=SHR,DSN=UPLOAD.WORK
//OUT DD DISP=SHR,DSN=a_F/U/V_dataset_name
//CONTROL DD *
LOAD
//

```

[0298] CONCLUSION of FIGS. 1-7

[0299] If two mainframe data centers have the program PCFORM, one data center can send any sequential dataset and a part or whole of a load library or a text library using

PC-based data transportation media. Any data center can be a sender or a receiver of mainframe data.

[0300] If the receiver does not have the program PCFORM, it can only receive a single load module, but it can not receive a library or other dataset. This feature is useful to send the program PCFORM to other mainframe data centers using PC-based data media.

[0301] Creation of Download Sequential Datasets

[0302] This invention solves the download and upload problems of a single load module, a part or whole of a load module library, a part or whole of a text library and a sequential dataset by converting to a download sequential dataset of fixed record format.

[0303] Fixed record format means that each record has the same length. As the explanation in the section of the Discussion of Related Art, only a sequential dataset of fixed record format can be downloaded to a PC and uploaded to a mainframe from a PC into its original form. A dataset which is not of fixed record format must be converted to a sequential dataset of fixed record format first to be delivered to another mainframe data center. The download sequential dataset of the sender and the upload sequential dataset of the receiver are defined as a fixed record format and their logical record length must be the same.

[0304] For undefined record format dataset, only the block size is 0.5 given and each block has only one record in it. So the block size is the maximum length of the records. Minimum record length can be 1 byte long. Each record can have different record length of from 1 byte to the maximum record length defined. This type of dataset cannot be reverted to its original form if downloaded to a PC and uploaded back to a mainframe. This type of dataset must first be converted to a sequential dataset of fixed record format to be delivered to another mainframe data center. A program load module library and its members, program load modules, are of undefined record format.

[0305] Many VSAM datasets and DB2 datasets are of undefined-type record format. For VSAM dataset, it's not called undefined record format, but it does have undefined record lengths. In many VSAM datasets, its maximum record length and average record length are given. Here, average record length is just an average value, not a minimum value. Each record length can be from 1 byte long to the maximum record length defined. When these VSAM datasets are copied to a sequential dataset, the sequential dataset must be of an undefined record format.

[0306] FIG. 1 shows that a single load module is delivered when the sender has the program PCFORM, but the receiver does not have the program PCFORM. The load module is an undefined record format dataset. In this case, the logical record length of the download dataset of fixed record length is defined with the same value as the block size of the library in where the single load module is located. Each input load module record of various length is padded with zeroes at the end of input data up to the block size length as in FIG. 19 and written out to the download sequential dataset. These padded records are uploaded and reverted to its original form in another mainframe using IBM utility programs IEBGENER and IEWL.

[0307] FIG. 2 shows that a part or whole of a load module library 0.5 is delivered to another mainframe data center

when both the sender and the receiver have the program PCFORM. As explained above, a load module library is of undefined record format. Only its block size is given and the length of one record can be different from other records and unpredictable. These undefined various length records must be converted to a sequential dataset of fixed length records. The conversion of a load module library is handled in two different ways in this invention. First method (FIG. 17) is: each input record is appended at the beginning of each logical record by a 4-bytes header which contains the real length of the undefined record and this record with 4-bytes header is divided into fixed length shorter records of download sequential dataset of fixed record length format. When uploading, the real record length in a 4-bytes header is analyzed first and proper number of following records are concatenated according to this length. The second method (FIG. 18) is: each input record of various length is first divided into shorter records of (the logical record length of the download sequential dataset minus 4) bytes length. Each divided record is appended at the beginning with a 4-bytes of header of 2-bytes of real data length and 2-bytes of sequence number of shorter records in the original input record. The first divided shorter record has the sequence number, 1. The last shorter record of each input record may have the data length shorter than the record length of the download sequential dataset of fixed record format. When upload, the upload record with the sequence number 1 in the header becomes the start of each output record and the real data length is in the header, so the concatenation is possible.

[0308] FIG. 3 shows when a part or whole of a text library is delivered to another mainframe data center when both the sender and receiver have the program PCFORM. In the case of a text library, all records are of fixed length and their length is 80-bytes long. No conversion is necessary because the output download sequential dataset is of fixed record format and its record length is defined as 80-bytes long. Each record from the input text library is written out to the output download dataset. Upload sequential dataset is also of fixed record format and record length is 80-bytes. Each upload record is written out to the destination text library. There is no conversion of text records.

[0309] FIG. 4 shows when a sequential dataset of fixed record format, undefined record format or variable record format is delivered to another mainframe data center when both the sender and the receiver have the program PCFORM. As mentioned above, for a sequential dataset of undefined record format, only the maximum record length is known, minimum record length can be 1 byte long and each record can have different record length. This is just the same situation as for the load module library. So, the undefined various length records are converted just the same ways as for the load module library records. The above two conversion methods of a load module library can be used for this sequential dataset of undefined record format.

[0310] Although the invention has been described with reference to a particular arrangement of events, features, sequences and the like, these are not intended to exhaust all possible arrangements or features, and indeed many other modifications and variations will be ascertainable to those of skill in the art.

1. A device for distributing a sequential dataset or a library of texts or load modules comprising:

- a sender programmer at the sender's site with a file transfer-enabled mainframe terminal for creating required JCL streams, for creating required datasets, for submitting required JCL streams, for creating the download sequential dataset records, for downloading the download sequential dataset to the PC and for creating PC-based data media to send to the receiver's site;
- a sequential dataset or a library at the sender's site to be delivered to the receiver's site;
- a download sequential dataset at the sender's site, which is of fixed length record format and its logical record length is 80 bytes or greater, for storing the output of the program PCFORM;
- a program PCFORM at the sender's site for converting the sequential dataset or library to the download sequential dataset of fixed length records;
- a downloaded PC hard disk file at the sender's site for storing the downloaded content of the mainframe download sequential dataset;
- a receiver programmer at the receiver's site with a mainframe PC terminal for receiving the data from the sender, for creating required JCL streams, for creating required datasets, for uploading the received data from the PC to the mainframe, and for submitting required JCL streams;
- an upload sequential dataset at the receiver's site, which is of fixed length record format and its logical record length is equal to the logical record length of the download sequential dataset at the sender's site, for storing the data uploaded from a PC at receiver's site;
- a destination sequential dataset or library, with the same record format as the record format of the respective dataset at the sender's site, for storing the data from the upload sequential dataset; and
- a program PCFORM at the receiver's site for converting the upload sequential dataset records to the destination sequential dataset or library.

2. The PC data media distributing device of claim 1, wherein said the program PCFORM logic can be expanded in various ways to convert any kind of dataset into said the download sequential dataset.

3. The PC data media distributing device of claim 1, wherein said the download sequential dataset at the sender's site can be reverted back to its original form at the sender's mainframe site.

4. The PC data media distributing device of claim 1, wherein, if a copy of said the download sequential dataset from the sender's site is delivered to the receiver's site, it can be reverted to its original form at the receiver's site.

5. The PC data media distributing device of claim 1, wherein said the downloaded PC hard disk file at the sender's site can be reverted to its original form if it is uploaded back to the sender's mainframe terminal.

6. The PC data media distributing device of claim 1, wherein, if a copy of said the downloaded PC hard disk file at the sender's site is delivered to the receiver's site, it can be reverted to its original form at the receiver's site, if it is uploaded to the receiver's site mainframe.

7. The PC data media distributing device of claim 1, wherein said the downloaded PC hard disk file at sender's site can be delivered to other mainframe data centers using PC-based data media, such as floppy diskettes, recordable CDs, various USB-port disks, ZIP disks, Internet world wide web download, Internet-based email attachment, TCP/IP FTP GET/PUT, and other PC data media available now and in the future, without the use of conventional magnetic tapes or tape cartridges, and can be reverted to its original form at other mainframe data centers.

8. The PC data media distributing device of claim 1, wherein said the downloaded PC hard disk files of any datasets at the sender's site can be used as backups at the sender's site.

9. The PC data media distributing device of claim 1, wherein the logic of said the program PCFORM can be included into other programs and get the same results.

10. A device for distributing a dataset comprising:

a sender programmer at the sender's site with a file transfer-enabled mainframe terminal for creating required JCL streams, for creating required datasets, for submitting required JCL streams, for creating the download sequential dataset records, for downloading the download sequential dataset to the PC and for creating PC-based data media to send to the receiver's site;

a mainframe dataset at the sender's site to be delivered to the receiver's site;

a download sequential dataset at the sender's site, which is of fixed length record format, for storing the output of the program PCFORM;

a program PCFORM at the sender's site for converting the dataset to the download sequential dataset of fixed length records;

a downloaded PC hard disk file at the sender's site for storing the downloaded content of the mainframe download sequential dataset;

a receiver programmer at the receiver's site with a mainframe PC terminal for receiving the data from the sender, for creating required JCL streams, for creating required datasets, for uploading the received data from the PC to the mainframe, and for submitting required JCL streams;

an upload sequential dataset at the receiver's site, which is of fixed length record format and its logical record length is equal to the logical record length of the

download sequential dataset at the sender's site, for storing the data uploaded from a PC at receiver's site;

a destination dataset, with the same record format as the dataset at the sender's site, for storing the data from the upload sequential dataset; and

a program PCFORM at the receiver's site for converting the upload sequential dataset records to the destination dataset.

11. The PC data media distributing device of claim 10, wherein said the program PCFORM logic can be expanded in various ways to convert any kind of dataset into said the download sequential dataset.

12. The PC data media distributing device of claim 10, wherein said the download sequential dataset at the sender's site can be reverted back to its original form at the sender's mainframe site.

13. The PC data media distributing device of claim 10, wherein, if a copy of said the download sequential dataset from the sender's site is delivered to the receiver's site, it can be reverted to its original form at the receiver's site.

14. The PC data media distributing device of claim 10, wherein said the downloaded PC hard disk file at the sender's site can be reverted to its original form if it is uploaded back to the sender's mainframe terminal.

15. The PC data media distributing device of claim 10, wherein, if a copy of said the downloaded PC hard disk file at the sender's site is delivered to the receiver's site, it can be reverted to its original form at the receiver's site, if it is uploaded to the receiver's site mainframe.

16. The PC data media distributing device of claim 10, wherein said the downloaded PC hard disk file at sender's site can be delivered to other mainframe data centers using PC-based data media, such as floppy diskettes, recordable CDs, various USB-port disks, ZIP disks, Internet world wide web download, Internet-based email attachment, TCP/IP FTP GET/PUT, and other PC data media available now and in the future, without the use of conventional magnetic tapes or tape cartridges, and can be reverted to its original form at other mainframe data centers.

17. The PC data media distributing device of claim 10, wherein said the downloaded PC hard disk files of any datasets at the sender's site can be used as backups at the sender's site.

18. The PC data media distributing device of claim 10, wherein the logic of said the program PCFORM can be included into other programs and get the same results.

* * * * *