DOCUMENT RESUME

ED 320 547 IR 014 462

AUTHOR Naden, Dave
TITLE Office of Impact Aid Computer System: Feasibility Study.
INSTITUTION Decision Resources Corp., Washington, DC.
SPONS AGENCY Department of Education, Washington, DC. Office of Planning, Budget, and Evaluation.
PUB DATE Mar 89
CONTRACT 300-86-0094
NOTE 139p.
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC06 Plus Postage.
DESCRIPTORS Comparative Analysis; *Computer System Design; Database Management Systems; *Data Processing; Feasibility Studies; Federal Government; *Information Systems; Local Area Networks; Microcomputers; Needs Assessment; *Public Agencies; Specifications
IDENTIFIERS Mainframe Computers

ABSTRACT

This feasibility study of the computer system of the Department of Education's Office of Impact Aid (OIA) was conducted to (1) assess the capabilities, as well as the drawbacks, of the current system and the other components of OIA's data processing activities; and (2) explore alternative computer systems, based on available technology and current needs of the program, which would enhance effective management and operation of the Impact Aid program. The first section of this report provides a description of the Office of Impact Aid and its current data system. The requirements for an alternative data system are discussed in the second section. The third section discusses the key factors that must be considered for a new system, and proposes two alternative systems, one a mainframe-based database management system, and the other a mainframe/microcomputer system with a local area network. Finally, the fourth section outlines the steps that should be followed in implementing an alternate computer system. Diagrams and attachments are appended to provide further clarification. (GL)

OFFICE OF IMPACT AID COMPUTER SYSTEM

Feasibility Study

Prepared Under Contract by:

Decision Resources Corporation
Washington, D.C.

Contract No. 300-86-0094

U.S. DEPARTMENT OF EDUCATION • OFFICE OF PLANNING, BUDGET & EVALUATION

Office of Impact Aid Computer System

Feasibility Study

By

Dave Naden

Submitted to:

Office of Planning, Budget and Evaluation

Contract No. 300-86-0094

March 1989

ACKNOWLEDGEMENTS

Numerous individuals contributed their time to the completion of this study.

From the Office of Impact Aid, Charles Hansen, Director, provided guidance on the overall needs of OIA, and Andy Larsen spent countless hours imparting his knowledge of the program and facilitating access to the other individuals who provided invaluable assistance in the gathering of information: Laurel Cornish, Veronica Edwards, Bob Farning, Marilyn Galvin, Jean Henry, Jim Link, Marco McElwayne, Bruce Reinhart, and Gregg Spencer, who also provided many critical review comments.

Information and review comments were also received by Ray Kudobeck and Bob Smith of the Financial Management Services.

From the Office of Information and Resource Management, Bob Bunting and Alex Poliakoff both offered help in explaining the current software system and hardware available to OIA.

Several persons from Pinkerton, Inc. also provided assistance in describing the technical characteristics of the Impact Aid data base system, including George Myrick, Joe Webber, and Bill Buchmeger.

From the Office of Planning, Budget and Evaluation, Nina Winkler (Planning and Evaluation Service) and Cathy Schagh (Budget Service) offered critical reviews of the documents. Special thanks go to the Project Officer, Joanne Wiggins (Planning and Evaluation Service), for her invaluable assistance during the course of this task order and her editorial comments and advice on the document at all stages of the study.

Finally, from Decision Resources, Jim Hutchinson provided technical consulting on systems analysis, Bill Lawson provided software expertise for drawing the flow diagrams, Bob Schrack wrote the description of the local area network and its component hardware and software, and Paul Walker provided further data base expertise. Adrienne Von Glatz produced the drafts of the document as they were written, and Susan Hennessy pulled the various pieces together into a coherent whole for final production. Saunders Freeland gave expert word processing for the production of the document. Finally, Babette Gutmann deserves special recognition for her tireless moral support, overall project direction, and editorial expertise.

TABLE OF CONTENTS

Contents (continued)

Contents (continued)

INTRODUCTION

In August 1988, Decision Resources Corporation (DRC) was asked, under Task Order #23, to conduct a feasibility study of the Office of Impact Aid's (OIA) computer system. OIA administers the disbursement of approximately $800 million a year to school districts under P.L. 81-874 and P.L. 81-815 and is responsible for verifying applications and information from school districts and for calculating payments. A computer system was put in place approximately 20 years ago to process the massive amounts of data needed by OIA to perform these functions. Since then, components have been added to improve the system; however, it is widely recognized amongst OIA staff that the current system has limitations which undermine the ability to process data efficiently.

DRC's role has been to assess the capabilities, as well as the drawbacks, of the current computer system and the other components of OIA's data processing activities, and to explore alternative computer systems, based on available technology and current needs of the program, which would enhance effective management and operation of the Impact Aid program.

Part I of this Feasibility Study, Office of Impact Aid Application Processing System, provides a description of the Office of Impact Aid and its current data sytem. Part II, System Requirements, discusses in detail the requirements for an alternative data system. Part III, System Alternatives, delves into the key factors which must be considered for a new system and proposes two alternative systems; finally, Part IV, Implementation Plan, outlines the steps that should be followed in implementing an alternate computer system. Diagrams and attachments are included in the Appendix to provide further clarification.

PART I - OFFICE OF IMPACT AID APPLICATION PROCESSING SYSTEM

This part of the document provides a description of the Office of Impact Aid (OIA) and its data system. Following overviews of the Impact Aid program and its data system, various components of the data system are discussed in detail, including transactions performed, machine-readable files used, computer-generated reports produced, and paper files maintained. To support the discussion and document listings, data-flow diagrams are included in the Appendix which illustrate how the data system operates.

OVERVIEW OF THE IMPACT AID PROGRAM

Impact Aid is a federal program designed to compensate school districts for loss of tax revenues resulting from the presence of federal properties in their districts and increased enrollments due to federal activities. It also provides districts with construction and disaster assistance.

Legislative Authority and Eligibility

Legislative authority and eligibility criteria for the program come from Sections 2, 3, and 7 of P.L. 81-874, as well as from P.L. 81-815. A district is eligible for assistance if it meets one of the following criteria:

- The district contains a significant amount of federally-owned property acquired since 1938 and suffers a substantial and continuing financial burden as a result (Section 2);
- The district contains a minimum number of school-age children who live on and/or whose parents work on federal property or are in the uniformed services (Section 3);
- The district has suffered substantial damage as part of a federally-declared disaster area (Section 7); or
- The district is eligible for construction assistance (P.L. 81-815).

Approximately 80 percent of Impact Aid funds are provided under Sections 3 and 2 of P.L. 81-874. The allocations to school districts are fairly consistent from year to year, whereas with Section 7 and construction assistance, the number of receiving

districts and the amounts received vary considerably from year to year. Approximately 3,000 out of 15,000 U.S. school districts receive some Impact Aid every year. These 3,000 districts are distributed across all 50 states, the District of Columbia, Puerto Rico, Guam, and the Virgin Islands; approximately 2,500 of them received aid the previous year from either Section 2 or Section 3.

Application Processing for Sections 2 and 3

Applications for assistance are processed by three branches of OIA, each with distinct but overlapping functions:

- The Field Operations Branch (FOB) conducts field reviews and verifications of data, including enrollments, expenditures, revenues, and tax rates;
- The Program Services Branch (PSB) is responsible for verifying that properties identified on applications are valid federal properties, and for keeping track of applications and payment batches; and
- The School Assistance Branch (SAB) is responsible for administering the payment process, helping verify the data on applications, and initiating payment actions.

Initiation

At the start of every school year the Field Operations Branch (FOB) sends a pre-printed application to each local education agency (LEA) that applied for aid the previous year under Section 2 or 3. This form contains a list of federal properties within the district that the LEA claimed the previous year. If the LEA believes it is eligible for assistance, it fills out the remaining information (additional property data and enrollment data for Section 3) and sends the application to the state education agency (SEA) for certification. The SEA then sends it to the Program Services Branch (PSB) of OIA. The PSB immediately dates the application; applications must be filed by January 31 in order for the LEA to receive aid for that fiscal year.

Initial Screening

The application is then processed by the three branches of OIA, each of which has continuing interaction with the other two throughout the process. The FOB receives a copy, keeps a log of all applications as they arrive, and maintains a permanent file of LEA data for field reviews, after validation by PSB and SAB. The FOB also creates a log of any new or non-continuing applications for a field review before payment. The SAB screens all applications for completeness (in particular, it verifies the local contribution rate), determines computer data consistency checks for the year, maintains the official permanent paper file, and resolves any problematic cases.

Property Validation

When the SAB is satisfied that the application is complete, the application is sent to the property division of PSB for property validation. Applications with problems are retained by either the property division or the SAB for further checking; others are put into the "ready for keypunch" queue. The PSB checks the property data and generates property IDs for newly claimed properties. During the course of the year, the property branch also validates all properties currently being claimed, using information from other government sources such as the General Services Administration (GSA) or the military services. They focus first on any new claims of federal property added to the year's application.

Computer Processing

When approximately 200 applications are ready, the SAB orders a batch to be sent to the Automated Data Processing (ADP) center for keypunching. The payments section of PSB prepares the batch for processing and actually sends it.

The computer performs calculations to determine eligibility and entitlements, based on three factors:

1) annual appropriation (and current allotment);

2) formulas in the computer code reflecting current legal entitlements; and

3) the data from the application regarding properties and school children for each LEA.

The computer generates the TR21 and TR22 forms for final visual inspection and approval by the SAB. These forms list the data necessary to compute entitlements for the year. The computer also generates several reports on the applications, particularly those found in error for various reasons, so that the SAB can resolve them. When all applications have gone through this process, there is a list of LEAs with maximum legal entitlements of federal Impact Aid payments (Sections 2 and 3) for a given fiscal year and a current payment calculation. This is the point at which the SAB may put a hold on an application for any reason.

Prepayment Processing

In addition to the determination of eligibility through applications under Sections 2 and 3, an LEA can be eligible for immediate payment through the prepayment system.

The prepayment system covers LEAs that were eligible under Section 2 or 3 and received payments the previous year and wish to receive payment prior to going through the full application process at the beginning of the new year. The LEA must have submitted a written request for a preliminary payment, which may be in the form of a brief letter. Starting with fiscal year 1989, the program statute calls for setting these payments between 50 and 75 percent of the previous year's total payments, depending on the type of district. These payments are deducted from later payments calculated on the basis of application data. The LEA is, of course, required to submit an application by the annual deadline. When all the current year data are finally collected, exact entitlements and payments are calculated, and corrections made for any deviations between current and previous payments.

Application Processing for Disaster and Construction Assistance

The disast assistance program and the construction assistance program are derived from authorizations separate from Sections 2 and 3, and an LEA may be eligible for payments under these programs at any time, independent of its eligibility for regular property-related payments. These two systems begin with a separate set of applications and validation procedures, including extensive on-site verification involving both the SEA and the FOB in the Office of Impact Aid, as well as other federal officials. When this process is finished, a list of all LEAs eligible for disaster payments and their entitlements is produced. These become entries in the routine batches (sets of payments) that are produced every week or so.

An important point with respect to the processing of applications for these programs is that the data used for calculations for disaster payments are gathered in the field. Final calculations, while complex, are performed manually in the disaster section of the OIA; only when the actual payment amount is determined are these data entered into the computer. In addition, the disaster and construction branches of OIA keep only paper files of all receipt control information and application data; none of this is automated.

Payment Processing

The payment process is triggered by an Advice of Allotment, which is a notice from the Office of Planning, Budget and Evaluation (OPBE) of the U.S. Department of Education (ED) that a certain portion of that year's funds are ready for the first set of payments to LEAs. This Advice of Allotment specifies a dollar amount in any of five categories: Section 2, Section 3(a), Section 3(b), disaster assistance, or construction assistance.

Prorating

At this juncture, the amount of money to be allocated for the entire year is known and the process of prorating payments based on entitlements and allocations has begun. An allowance is made to account for overpayments, errors, and special categories of payments; therefore, the total appropriation for a fiscal year is never completely allocated during the year, nor is it all available at the start of the year. For example, if an LEA is determined eligible for one million dollars, it will receive two or three payments throughout the year, with perhaps 50 to 75 percent of its full entitlement for the first payment. As the year proceeds, more information becomes available to OIA, data on prepayment LEAs arrive, and previously made errors are found. Corrections are continuously being made to some of the LEAs' legal entitlements, which change the overall relative entitlements to LEAs. These corrections are instigated by the SAB and changed via transactions in the computer system. Because of this, the prorating process occurs several times during the payment year, finely tuning the relative amounts allocated to the LEAs, while never going above the appropriation for that fiscal year for each category of payment.

Payment Batching

For initial payments, data from the applications are keypunched. ADP then generates a set of reports on each LEA on the list, which contains all the application data. Then both the SAB and PSB, in close communication, perform a series of final checks, mostly by hand, using the lists of applicants generated by ADP. For every LEA, a decision is made as to whether to go forward with a payment action or hold. An LEA may be put on the hold list for any reason the SAB deems appropriate, such as data that do not look accurate or previous knowledge about a problem. Payments are then generated for those LEAs not on hold. For those LEAs with problems, the TR21 form is held for correction in the SAB until the next batch or until data on the application can

be validated. Follow-up payment batches are prepared by the SAB, consisting of TR21 forms that have been corrected.

For this batching process, the PSB prepares several record sheets and lists for later processing and checking. These include a hash total record (a randomly generated but unique number), a list of TR21s and TR22s, a batch number sheet, fiscal year totals, and a voucher recording the payment action. This information all goes into a batch request, which is sent to ADP for processing. A typical batch consists of approximately 250 actual cases, each case normally being one payment action for one LEA. Batches are produced at roughly one per week throughout the year. The 250 cases are chosen from the lists of eligible LEAs, such that they are distributed evenly throughout states and the nation.

With every batch, the PSB updates the daily control log, generates special vouchers to indicate special payments, if necessary, and requests Congressional letters (first payment of the year only).

Transfer of Funds

The PSB requests the Financial Management Service (FMS) to make the payments, and requests ADP to produce the payment disk pack, which is used by the FMS to generate the electronic transfer of funds to the LEAs. When the computer system processes these payment transactions, it also updates several of its files, which keep track of the current status of all the applications, overall accounting information, and payments made. The FMS makes the transfer of funds to the banks where the LEAs have accounts, and sends the payment vouchers to the LEAs, informing them of payments processing and methods of calculation. The vouchers closely resemble the TR21 forms mentioned earlier, except that they contain the complete payment calculation.

Audits and Field Reviews

All new applicants receive a detailed audit from the FOB; others receive audits according to priority lists which are based on the amount of money obligated and data at critical cutoff points. The audit consists of a site visit by an officer of OIA and the verification of all data claimed on the most recent applications, including properties, numbers of children in various categories, local contribution rates, and total current expenditures. When errors or anomalies are found between these data and current records, the main files are corrected. If this causes any previous or current payment to be rendered inaccurate, the corrections are entered into the computer files, and future computer transactions take this change into account in calculating payments. If overpayments are found, the amount is stored in the accounts receivable file and deducted against future payments. The past five years of payment records are kept available for this type of correction. After six years, cases are considered closed, meaning that payments made more than six years ago can no longer be adjusted or corrected.

For new or non-continuing applicants, a field review is automatically performed before any payments can be made. The SAB has a list of all new applicants and puts them on hold along with other problematic applications until it receives a validated field report. The SAB checks the field report by hand and if the data appear correct, the TR21 is submitted in a batch to generate a payment. If there are questions about a field report, any suspicious data are re-verified by the FOB, and the TR21 is corrected by hand in the SAB and submitted as part of a follow-up batch of payments.

Final Payments

One other type of payment is generated near the end of each fiscal year, known as a spread payment. Its purpose is to disburse whatever funds remain for that fiscal year's allocation that are not needed for special provisions. For those LEAs that have

not been paid their full legal entitlement for the year, a final prorating is done using the funds now available and the total amount of remaining entitlement not received by LEAs. New percentages are computed by hand, entered into the computer code which generates the payments, and applied to these LEAs for final payments.

OVERVIEW OF THE DATA SYSTEM

The data system has four major components:

- o transactions on the computer files;
- o machine-readable files;
- o computer generated reports; and
- o paper files.

Underlying these components is the actual computer system in place to process applications and payments.

The computer system:

- o keeps a permanent and accessible record of all LEAs who have ever applied for aid, including information about their applications, dispositions, and payments;
- o allows careful scrutiny and control of applications and their processing by various persons in the OIA through the use of an on-line receipt control system with several checkpoints;
- o enables and enhances OIA staff's ability to validate claims of properties, students, and other data on applications;
- o enters the necessary data and performs calculations of eligibility, entitlements, prorated entitlements, and payments for all applicants; and
- o generates the tape or disk files necessary for the FMS to process payments to LEAs.

The computerized system used by OIA is a complex set of COBOL programs and large data files. These files are kept on-line where possible, and often include the last six years of data; in other cases, only the current year is on-line and the remainder of the data is stored on tapes. The system is operated via a set of predetermined

transactions, each with an identifying code, and its own function or set of functions. The transactions perform all the routine functions needed to operate the system, such as allocating money for disbursement, computing actual payments, generating reports, entering and modifying the application data, and storing records of transactions.

Associated with these transactions are several input data forms (keypunch forms), as well as many reports indicating the current status or history of recent actior Reports are generated in three circumstances: 1) to indicate the status and errors from a set of transactions; 2) at predetermined periodic intervals (i.e. weekly or monthly); or 3) on request from the OIA when needed. The system must be accessed entirely through batch processing, as there is no interface allowing immediate access. This is done via a standard work request form completed by the OIA, added to the actual input data forms (if needed), and sent to ADP for execution. Most types of work requests can be ordered at any time; some requests depend on others having been performed first (for example, payments cannot be generated until money has been allotted).

All of the transactions described here are performed through a contractor, who receives a work request and input forms, if appropriate, from the OIA. Out of approximately 70 persons on the OIA staff, fewer than 10 have any contact at all with the current computer system, and direct computer interfacing is done exclusively by the contractor. The involvement of OIA staff consists primarily of preparing input forms: the division chief of the SAB prepares TR21 and TR22 correction forms; the division chief of the FOB prepares field report data correction forms; the property analysts prepare property correction input forms; and an individual in the PSB prepares ID file update data entry forms. In addition, the division chief of the SAB works out annual parameters (such as funding limits and definitional changes resulting from the law) and edit checks with the contractor, who then develops the appropriate computer code. For all of the above, the payments section of the PSB prepares batch header sheets, assembles input forms for the batches, and submits them to the contractor. The contractor, in turn,

sends reports back to the appropriate branches in the OIA, indicating the results of the transactions.

While the system maintains data for six fiscal years, most transactions relate to the current fiscal year. Before a new fiscal year's payments can begin, several processes must be initiated by the OIA to initialize the computer files and set up the system for a new year:

1) The entire data base is backed up to tape;

2) The data for the sixth year back is removed to make room for the new year;

3) The new appropriations legislation must be interpreted by OIA, discussed with ADP, and translated into the part of the computer code that calculates entitlements and payments;

4) The annual appropriation must be entered into the computer;

5) The Property 2 and Receipt Control files (see next section) must be initialized (set to zero);

6) Pre-printed applications, 5-cards, TR2 cards, and receipt control listings must be generated based on the previous year's recipients; and

7) Any edit checks desired for the current year must be created by OIA and given to ADP to enter into the computer code.

These processes should be finished by the end of January of the fiscal year in question.

SYSTEM COMPONENTS

The major components of the system are a set of prescribed transactions and associated input forms, the computer files created or affected by the transactions, the outputs from them in the form of reports or other documents, and various paper files used by the OIA.

The inventory of transactions is grouped similarly to the computer files. In addition, the necessary ordering of the transactions within a fiscal year is provided. The description of the major data files includes the kinds of variables they contain, their approximate size, and their important key fields. An inventory of reports and other

output is provided, grouped functionally so that it can be directly related to the transactions performed. Finally, an inventory of the important paper files kept by the OIA is provided, including a description of each, to provide a larger picture of the data processing necessary to operate the office, regardless of whether particular components are automated or not. If there is a direct relationship between given paper files and computer files, this is provided in the description as well.

The descriptions of each major component are accompanied by four data flow diagrams located in the Appendix. *Attachment A, Application to Payment Processing*, illustrates the functions of the various components of the OIA in the processing of the applications. The diagram begins with the generation, in September-October, of the pre-printed applications for the new year, and ends with the payment tapes prepared for the FMS. Two subgroups of this process are illustrated by the following data flow diagrams: *Attachment B, Field Report Processing*, illustrates the somewhat separate function of the Field Office, and *Attachment C, Management of ID File*, shows the validation of the LEA IDs in relation to overall application processing. Finally, *Attachment D, Transactions and Computer Files*, illustrates the relationship between the transactions and the computer files.

All these diagrams use standard structured design techniques: boxes or rectangles indicate documents or real data--input forms, paper logs, computer files, IBM keypunch cards, applications, reports, etc. Circles indicate some process or action performed. Data flow is indicated by arrows connecting figures. Data forms that refer to specific reports, input forms, or computer files are given their code number or name, referred to in the lists provided. Where diagrams do not fit onto one page, arrows with matching letters are used so that the connections can be easily seen.

Transactions on the Computer Files

The transactions described in this section are grouped by the four main file systems--Property 1 System, Property 2 System, Receipt Control System, and Payments and Entitlements System. Most transactions may take place at any time of the year, irrespective of the timing of applications; however, *those preceded by a number must necessarily take place in the order indicated for each fiscal year.* There is one important exception to this ordering: for pre-payments, step 4 (Initial Payment processing) is not necessary, as these payments are based on payments for the prior year, not the current year's application.

Property 1 System

Insert, delete, or change data transaction - This is used to add entire new records, delete entire records, or change any fields on existing records in the file. It uses forms 631 and 632 and is performed approximately 200 times per year.

Insert, delete, or change additional data transaction - This allows changes, deletions, or additions in the county codes and names to any property record in the file. It uses input form 641, 642, or 643 and is performed approximately 200 times per year.

Property 2 System

Edit/update transaction (DHPP3000) - This is the basic method of entering the application data into the Property 2 file. If an application does not pass the edit check, it is entirely rejected; if it passes the edit but not the update check, the data are entered, and the property validation report indicates the problem(s). The data are keyed onto the input form 709 from the RSF-1 (the application), then entered into the system. Approximately 3,000 applications are entered per year, 200 of these entirely new. Approximately 1,000 applicants request preliminary payments first, and the entire application is entered later.

Property change - This transaction only changes information about a particular property claimed on an application. It uses input form 710 and is run approximately 200 times per year.

Sec 3 add, delete, or change - This transaction is used to change, add, or delete information from the Property 2 file, including an entire application. It uses input form 711 and is performed approximately 200 times per year.

Sec 2 add, delete, or change (712) - This transaction is the same as the previous one, but for Section 2 only. It is used perhaps 50 times per year.

Receipt Control System

[2.] *Annual start-up* - This is performed once per year, after annual file initialization, and consists of preparation for the coming fiscal year. On request, the system produces checkpoint punch cards (5-cards and TR2 cards), pre-printed applications, and a printed list of the receipt control file listing these applicants for paper records kept in the FOB. To do this, a program reads the previous year's payment file to determine which LEAs received payments and initializes the new year's receipt control files. This transaction requires no input form, only a work request.

[3.] *Checkpoint transactions* - These are the 5-cards (from FOB) and TR2 cards (from PSB) indicating receipt and approval of incoming applications. They update the receipt control file, and flag whether an LEA may receive a payment or not. One card of each type is entered for every continuing LEA, or approximately 2,500 per year each. They are the same cards generated in the annual start-up process.

[4.] *Initial payment processing* - This transaction is performed after the application data are entered into the Property 2 system. It reads the application data, checks the receipt control file for the go-ahead flags, and computes the TR21 data, which is then ready for input into the payment system, exactly as TR21 transactions normally are. This transaction is performed once per year for each LEA applying, or about 3,000 per year. There is no input form, only a work request.

Payments and Entitlements System

[1.] *Annual file initialization* - This transaction is performed once a year, before new transactions can be started for a new fiscal year. The transaction archives LEA records over five years old, initializes the ID, accounts receivable, and main and state control ledger files. It uses no input form, only a work request.

[5.] *Initial allotment transaction* (01) - This establishes the annual allocation for each component of Sections 2, 3, and 7 (disaster), and construction. It occurs once a year only. This transaction is necessary before any payment processing may begin for a given fiscal year. It initializes the accounts receivable file, and the main and state control ledgers. It uses a special TR1 input form, one page long.

[6.] *LEA statistical transactions* (21) - This initiates estimates and establishes obligations and/or payments for Section 3. When initial payment processing is performed, a TR21 form is generated, reflecting the current data for the applicant. After this point, the same form is used by OIA to make corrections and generate further payments throughout the year. Since each LEA receives approximately three payment actions per year, this transaction is performed approximately 10,000 times per year.

Batch Header Transaction - This is a header sheet containing such information as the batch number, hash totals, and number of transactions.

Amend allotment transaction (02) - This adjusts the annual overall allocation after the initial amount has been established. Normally performed only once or twice per year, it changes the amounts in the accounts receivable file, and the main and state control ledgers. This transaction also has a special TR2 input form.

Change to applicant accounts receivable transaction (07) - This is used to increase or decrease the accounts receivable file for an LEA receiving disaster payments, and is rarely used. It alters the total in the accounts receivable file and uses the TR7 input form.

Identification transaction (12) - This establishes or updates the ID file for any applicant. It is used perhaps 100 times per year to enter a new applicant, and perhaps 100 to 200 times to change information on existing applicants. It uses the TR12 form (1 page) for entering new IDs.

Cancel payment transaction (14) - This cancels the most recent payment record on the statistical file for any LEA; it also adjusts state and main control files. This is used to stop payment just before the payment disk pack is generated if the SAB finds any reason to do so. It uses the TR14 input form.

Refund transaction (16) - This adjusts the applicant accounts receivable file, due to receipt of an amount owed by an applicant. It is rarely used because normally any money an applicant owes is offset against future payments. It uses the TR16 input form.

Section 2 statistical transaction (22) - This initiates estimates and establishes obligations and computer payments for Section 2. It works the same as the TR21 transaction, except that it contains far less data (only property) and is far less frequent. It is performed approximately 1,000 times per year. It uses the TR22 form, similar to but simpler than the TR21.

DIS (disaster) statistical transaction (28) - Similar to the above, this is for disaster payments only. The calculations of disaster payments are done entirely manually; therefore, this transaction is very simple, consisting only of entering a dollar amount. It is performed from 10 to 100 or more times per year. It adds records to the DIS payment file, adjusts the accounts receivable file if necessary, and adjusts the main and state control ledgers. It uses the TR28 form, which contains only identifiers and the dollar amount of the disaster payment.

Closed years change transaction (35) - This adjusts the balance of obligation amounts on main and state control ledger files for non-active years. It is used only occasionally and requires a special TR35 input form.

Field report transaction - This transaction is performed whenever a batch of field reports is received and prepared by the FOB, approximately 2,000 per year. The data entered updates the receipt control file indicating that a field report has been performed, and updates altered data on child counts, etc. in the payment files (LEA, disaster, or Section 2). The FOB uses two pages of pertinent data from the actual field report for the input form.

Percent update transaction - This is performed to reset the proration and payment percentages used to compute LEAs' payment amounts. It changes the control record of the ID file for the desired year. This transaction is performed at most a few times per year, particularly when calculating spread payments. There is a special input form for this transaction.

Congressional names edit/update - This transaction is used to add new names to the Congressional file for each district. Names can be updated whenever necessary, but normally they are deleted and added every election year. A special input form is used, with all necessary fields.

Machine-Readable Files

Machine-readable files are in four groups, roughly corresponding to their function: 1) the Property 1 system contains property data for all federal properties; 2) the Property 2 system contains the annual application data; 3) the Receipt Control system contains files reflecting the current status of every application; and 4) the Payment system, a group of files, is used for all eligibility and payment calculations and storing data pertaining to the same.

Property 1 System

Property 1 file - This is a permanent record of all federal properties ever claimed in the Impact Aid program. All properties currently being claimed are validated every year (verified by an outside source). The file contains Section 2 and Section 3

properties. Properties generally are validated or updated during the summer. This file, then, contains the most recent status on all properties, but no history. Once a year, a master list is generated from this file, containing every eligible federal property for each state, whether currently being claimed or not. All LEAs receiving Impact Aid receive this list, from which they choose properties to claim. When the applications are received, during January-February, any new properties claimed on the application (that is, not on the current master list) are assigned a property ID and entered into the permanent Property 1 file. They are verified later in the year. The file contains the 13-digit property ID, consisting of the state (2 digits), county (3 digits), federal agency that has jurisdiction over the property (4 digits), and a unique 4-digit number generated by OIA. The 4-digit federal agency code is significant, as it is used in calculating child counts by category (i.e., type of federal property). The file also contains such things as the name of property, agency, address, city, state, acreage, and year first claimed.

Number of variables:	approximately 40
Number of observations:	approximately 40,000
Key fields:	state, unique property ID

Property 2 System

Property 2 file - This is a subsystem consisting of all the application data for each year and the data reflected on the TR21 form, which are calculated by aggregating and categorizing child counts from the application. Only the current year's data are kept on-line; data for the previous years are maintained on tape. This file also contains the ID's used in the Property 1 System (mentioned above). It also has the unique application number *for that year*, which is the unique LEA identifier (see ID fi. below) plus the fiscal year field. Note that this file also contains the EIN number (see below), which uniquely identifies every LEA as a legal entity, regardless of year. The data here are hand verified by OIA for accuracy of keying, using the TR21 forms. This TR21 data then feeds into the Receipt Control and Payment Systems (following sections), where it is

used to compute determinations of eligibility, obligation status, entitlements, prorated entitlements, and payments.

Number of variables:	approximately 200
Number of observations:	approximately 3,000 (per year on-line)
Key fields:	FY, application number (11 digit), state

Receipt Control System

This system consists of one file, containing information on every application and its current status. Fields include the number of children approved, whether there has been a field review, what sections of the law apply, most recent payment code, date the application was received by FOB, and date approved by the property division and SAB. The past five years are kept on-line. (Note that records of disaster payments are not in this file.)

Number of variables:	approximately 100
Number of observations:	approximately 18,000
Key fields:	FY, application number, state, program type (Section 2 or Section 3)

Payment System

The payment system consists of eight different files, some of which are permanent and relatively fixed, others of which are updated every time transactions take place.

Main control ledger - This is an on-line checkbook containing the national totals, to date, for each of the three appropriations: Section 2, Section 3, and disaster assistance. Variables include fiscal year, program type, total appropriation, total obligation, total payments for this batch, and the allotted but unobligated balance. Every time a batch of payments goes through the system, a new record is added to this file, approximately 50 to 100 times per year. Six years of data are kept on-line.

Number of variables:	approximately 15
Number of observations:	approximately 500-600
Key fields:	FY, project type, reference number (batch number)
Note:	1 observation/batch/program type

State control ledger - The same as the above, except it is broken down by state, and a state code is included.

Number of variables:	approximately 15
Number of observations:	approximately 25,000
Key fields:	State, FY, program type, reference number
Note:	1 observation/batch/state/program type

Accounts receivable ledger - This contains records for LEAs receiving overpayments, and includes debit, credit, total balance due, and maximum collection percentage. When an LEA is found to have received an overpayment, an entry is made here. As it is paid off or deducted from future payments, the balance due is reduced accordingly. Six years of data are on-line.

Number of variables:	18
Number of observations:	approximately 1,000 total (6 years)
Key fields:	applicant number, state, FY, program type
Note:	maximum records would be 1/LEA/year

State accounts receivable ledger - As above, but this has only IDs and total dollars that all LEAs have been overpaid, by state for each year.

Number of variables:	6
Number of observations:	maximum of 300
Key fields:	FY, state, program type
Note:	maximum records would be 1/state/year

Congressional file - This contains each Congressional district, all LEAs within it, and the name and address of the relevant Senators and Representatives. It is updated every two years, or as needed.

Number of variables:	6
Number of observations:	approximately 535
Key fields:	state, Congressional district code

ID file - This is a complete list of all LEAs that have *ever* received Impact Aid, Section 2, Section 3, or disaster assistance. It has each LEA's unique 11-digit ID, assigned when first applying, which consists of state code, fiscal year, section applied for, and a unique 4-digit number. Since LEAs can apply in multiple years, the FY part of the ID changes; the other parts of the ID serve as a unique LEA ID across years.

There is another identifier, called an employer identification number (EIN), which is a federally-assigned 12-digit number, uniquely identifying every *employer* in the U.S. (like a social security number). The ID file also contains name, address, first year applied for Impact Aid, and Congressional district. In addition, there are control fields on the ID file, used throughout the year for processing purposes, containing such things as status of the LEA for the past five years. There are also data on current prorating percentages from the TR21 forms. The records on this file are of three types: 1) basic ID information; 2) control records, one per year; and 3) a master record containing current processing controls and information.

Number of variables:	approximately 40-200
Number of observations:	approximately 6,000 active, 18,000 total
Key fields:	application number, EIN number, FY last applied
Note:	this file has three record types, and is hierarchical

Payment files, including *LEA* (Section 3), *SEC 2*, and *DIS* (disaster) - These files contain the payment transaction data, including all the data from the TR21 (average daily attendance (ADA), local contribution rate (LCR), membership, child counts, proration percentages, and entitlements for every category), and payment amounts for up to 10 payments per year. The disaster assistance and Section 2 files are quite small, while the LEA file records are 1,800 characters long. This file grows rapidly each fiscal year, having a new record added for every LEA and every transaction, for each year. All the data are repeated for each transaction in a year to accommodate occasional changes in the data. This file allows a complete accounting of how every payment was calculated.

Number of variables:	LEA, approximately 250; disaster assistance and Section 2, approximately 25 each
Number of observations:	each year beg: mpty, ends up with 1 observation/LEA/. ansaction, or approximately 50,000 (2,500 LEAs x 3 transactions x 6 years)
Key fields:	Application number, state, FY, project type

CRS file - Central Registry System is a crosswalk file (a file which translates identifiers from one set of codes to another) containing identifiers and is used by the

FMS for other purposes. It contains a code assigned to all entities receiving any dollars from ED, and allows EIN codes (see ID files above) to be linked to these entities. It also allows communication between disaster assistance applicants and the rest of the payment system.

Computer-Generated Reports

There are several types of computer-generated reports: 1) those associated with specific types of processing, generally batch processing, whose main purpose is to indicate the status of transactions; 2) those ordinarily issued at pre-defined intervals irrespective of batch processing; and 3) those issued whenever requested. In addition, many reports generated are one-of-a-kind, as requested or needed by the OIA. There are approximately 100 such reports requested per year; most are listed in Attachment E.

Paper Files Used in the OIA

Numerous paper files are maintained by the OIA. Those that are key to the daily maintenance and upkeep of the system of payments include:

- o accounts receivable ledger - done by hand, overpayments only and collections against them. This ledger should reflect the computer accounts receivable file, except that it is more up-to-date (kept in SAB);
- o record of edit windows sent to ADP - prepared by hand, first produced at the start of a fiscal year, but modified perhaps 10 times per year. The current windows are embodied in the computer code and changed when OIA requests (kept in SAB);
- o official case file - contains copies of all data on every applicant, i.e., the application, field report, TR21 form, and any other documentation pertaining to the LEA. This file contains the last six years of records (kept in SAB);
- o daily control log - a record of all batch requests (kept by PSB, payments), with such information as batch number, number of TR21s and date. This should be reflected in the main control ledger computer file;
- o batch file - containing batch printouts for a paper record of batches processed, for quick access or reference, kept perhaps 2 to 3 months only;

- disaster payment log - a record of all disaster payments by fiscal year, including the entire application. This is similar to the official case file (above) and serves as such for disaster payments (kept in FOB, disaster section);

- construction payment log - record of all construction applications, and a receipt control-type log of all actions pertaining to these applications, including dollars awarded, if any (kept in construction branch). This is the official case file for construction payments;

- payment memoranda file - a record of specific requests from SAB to PSB regarding payments (kept in payments branch of PSB);

- receipt control log - a record of every application received, date received by FOB, by property, data keyed in system, first payment code, sections of the law applied for. Generated at start of year to match preprinted applications sent out, new IDs added as they come in. This serves as a check against the computer file receipt control log, which should reflect its contents (kept in FOB);

- list of requests for new applications and application information from LEAs - used for sending out new applications at start of the new fiscal year (kept in FOB);

- permanent copy of all field reports (one in FOB, another in official case file, above) - the date of the field report should be reflected in the receipt control file in the computer and the data changed as a result of the field report should be reflected in the payment files;

- log of all requests sent to ADP for updating files, based on new field reports - serves as a check against errors made in entering field report transactions, as well as against the receipt control file (kept in FOB); and

- property documentation supporting all properties ever claimed in Impact Aid program - used for reference in the property validation process. Only a fraction of the data here is reflected in the computer Property 1 file (kept in property section of PSB).

PART II - SYSTEM REQUIREMENTS

This component of the Feasibility Study discusses the requirements for an alternative data system for the Office of Impact Aid. In designing a new system, there are two prerequisites:

- o all of the functional abilities of the current OIA data system must be maintained; and
- o new capabilities and improvements that respond to the identified problems and limitations of the current system must be included.

The current data system used by OIA was described in detail in the first part of this document. Its capabilities are summarized below to set the context for the consideration of the requirements that must be maintained in a new system. A discussion of problems and limitations of the current system is then presented, followed by a description of additional capabilities to respond to these problems that will be considered for inclusion in a new system. The next section provides an overview of the hardware/software/personnel requirements for a new system. A final section summarizes some issues that are important to keep in mind when identifying system requirements and when designing a new system.

In addition, seven attachments in the Appendix provide more detail: *Attachment F, Logical Data Groups With Keys; Attachment G, Calculation of File Sizes for the Impact Aid System; Attachment H, Variables Needed for Application and Payment Data; Attachment I, Transactions and Computer Files. New Schema; Attachment J, Example of Interactive Data Corrections; Attachment K, Schematic of Batch Processes; and Attachment L, Description of Transactions with New OIA System.*

CAPABILITIES OF THE CURRENT SYSTEM

Maintaining current capabilities is a key component in the design of a new system. An examination of the functions OIA performs and what the current computer system does to support these functions shows the following current capabilities:

- Data storage of all the files or their functional equivalent, including all necessary fields, as listed in Part I, "Machine-Readable Files" (p. 17). The overall system of files has the capacity to grow slowly over a 5 to 10 year period; this size currently increases an estimated 5 percent per year.

- Data retrieval, update, and deletion in batch mode of the above files, as described in Part I, "Transactions on the Computer Files" (p. 12-17), with a 1 to 2 day turnaround, depending on the transaction. The system also enables the user to control the ordering of the start-up transactions (#1 through #6), as indicated on that list.

- Report generation, of the variety and frequency specified in *Attachment E, Computer-Generated Reports*, of a reasonable print quality, again with a 1 to 2 day turnaround time. This includes the payment vouchers when payments are generated.

- Input of new annual data from the applications, the ID file, the Property 1 file, and the correction forms (TR21, TR22) such that extensive computer edit checks, as specified by the OIA, are performed on data values, ranges, and legal identifiers.

- Storage of computer code used to perform routine batch and interactive processing; modification of the programs that process the above files at OIA's request to reflect changes in appropriation amounts, percentages, and entitlement calculations every year; maintenance of a history of these modifications; and the ability to maintain computer programs reflecting six years of differences in the law.

- Hand and visual checking, modification, and verification of application and field report data by several sections of the OIA (PSB to FOB to SAB to property to payments), including the ability of OIA personnel to closely monitor and control the application processing, and, in particular, the ability to hold a payment right up to the creation of the payment disk pack for the FMS.

- Automatic generation of letters to LEAs and Congressional districts as follows: to members of Congress upon first fiscal year payment; less than $5,000 rejection letters to LEAs at appropriate times; notice of receipt of application letter to LEAs when received by FOB.

- The ability to create computer files (obligation tape and payment disk pack) which are usable by the FMS of ED to generate the payments.

- Generation of punch cards, pre-printed applications, and receipt control listings every year for hand checking applications and controlling the process of approving an application for payment.

- Regular backup and retrieval of system files and programs for security.

- Flexibility in the system that permits the design and production of new and ad hoc reports.

- The ability to save data pertinent to the computation of payments for at least a five year period for legal purposes. This includes a history of the data that are now in the payment file(s), including whatever changes are made during that time via the TR21 form or field reports. It also includes property data (specific to an application), Tables 1 through 10 (application), TR21 and/or TR22 data, and actual calculated payment data.

LIMITATIONS OF THE CURRENT SYSTEM

Determining requirements for a new OIA computer system depends on identifying problems and limitations with the current system. These limitations include the following:

- The current system is much too slow.

- The current system allows inconsistencies in the data, that is, changes made in one place are not reflected throughout the system.

- OIA staff do not have enough direct access to the data base, and must work through a contractor for even the simplest requests.

- Generating new reports or performing analyses on the current data base is extremely difficult, and in many cases, impossible.

- Many aspects of the OIA's operations could benefit from computer support where none now exists.

In short, the system is not automated enough, is not integrated, and is not accessible to many staff who have regular need for access to the data, but have little computer knowledge.

CAPABILITIES OF A NEW SYSTEM

An improved system should incorporate the following:

- The data base should be reorganized to ensure efficiency and simplicity.
- Transactions should be redesigned to incorporate programs which ensure file and data consistency when transactions are performed that alter application and/or payment data.
- Several transactions currently performed in batch should be redesigned to be done more interactively to ensure faster turn-around.
- The use of batch processing should be maintained to allow for on-line record-keeping, and to maintain compatibility with the FMS.
- New essential features should be added, including the automation of several processes which are currently done manually and the maintenance on-line of some of the currently-maintained paper files to ensure responsiveness to the identified problems and limitations of the current system.
- New desirable (but not essential) features might be added to enhance OIA's abilities to administer the Impact Aid program.

The Data Base

Reorganizing the data base will: (1) help eliminate redundant data; (2) make quality control within the data base more manageable; (3) reduce the overall size of the data base and increase the speed of access to the data base; (4) make the data base easier to understand and manipulate; (5) make the implementation of a new system easier; and (6) make possible a far greater range of access and analytic capability on the data base. With the proposed schema, every current transaction will be possible, and, in most cases,

will execute far more quickly. In addition, many new ways of manipulating the data base, which are not currently possible, will become routine.

Major structural changes proposed in the data base are:

- integration of *applicant* data, *payment* data, *receipt control* data, and *accounts receivable* data. These are all unique within a fiscal year and LEA, and logically belong together;
- consolidation of all state-level data within each year, including *state control ledger* and *state accounts receivable*;
- addition of a file with annual constants and parameters, so that manipulations are easier. These include: appropriation by program type, current edit checks, and prorating percentages; and
- addition of an *application and payment history file*, in order to reduce the size and increase accessibility to the current *application and payment files.*

Relevant to this discussion are four attachments in the Appendix: *Attachment F, Logical Data Groups With Keys*, gives a complete description of the newly organized data base at the highest conceptual level. Following this, *Attachment G, Calculation of File Sizes*, shows exactly how the new data base would be laid out and a rough approximation of access frequencies. Since the application and payment data form the core of the Impact Aid system, *Attachment H, Variables Needed for Application and Payment Data*, provides more detail on these files.

Transactions and Changes to the Data Base

Throughout the discussion of the forthcoming sections, the reader may want to refer to *Attachment I, Transactions and Computer Files, New Schema.*

One of the benefits of the suggested data base reorganization is improved internal consistency--a given data item is represented only once in the data base, and if this item is changed, that change will be permanent and final. If a history is needed, as with application and payment files, it will, of course, be saved first.

There are, furthermore, several assumptions that must be made regarding changes to the data base, the ordering of these changes, and the implications for data integrity. The assumptions are:

- Changes made to the permanent ID file will be made before applications are entered; for a given fiscal year LEAs whose applications precede the cutoff date will have their ID added to the file.

- New properties will be added to the property file before applications claiming them can be processed.

- Data found on the state-level file and the batch payment file (containing the main control ledgers) reflect occurrences during transactions. Therefore, it does not make sense to manually change data in these fields, and such changes should not be allowed.

- Data entered into the new *field report log file* should be reflected in the main application and payment files(s) within a short period of time, as these data are needed to update the main files. If an historical record is needed, the receipt control fields will indicate the date of the field report and the "history" file will contain data used before that field report was applied.

- There is an implied agreement between the annual appropriation amounts, the main control ledger, the state control ledgers, and actual payment records for each LEA. Therefore, if this appropriation is amended, adjustments must be made to the latter two files. Further, the running totals of dollars obligated and paid, separated by section (CAN #), should agree with LEA and state totals at all times.

- Key fields should not be allowed to take on null values or have duplicates; this includes primary keys for data base files (see *Attachment F*), or foreign keys (keys which are not used to identify the record in question, but to refer to records in other data base files). These include applicant ID, property ID, fiscal year, state, EIN number, and Congressional district.

- If changes are made to an applicant's data for a given fiscal year, all other fields that are a function of those changed fields or went into the original computation of the changed fields will also be changed on that same year's data for that applicant. This specifically refers to application data with properties and child counts.

It should be immediately noted that these restrictions are far from complete; they are listed here because they are invariable. Extensive edit checks are also incorporated into

all of the transactions which involve new data or updating data; these are discussed elsewhere. Most of these assumptions are tested at the time the transactions take place. However, it is standard practice on a data base of this size to provide several small programs whose sole purpose is to insure the data consistency and integrity of the data base where the data base management system software itself does not provide this type of checking.

The last assumption listed above is particularly important because it speaks to a recurring problem with the OIA's data base. The problem is best understood by separating the components of the group of files called application and payment files (see *Attachment H*). Once an LEA becomes an official applicant in a given fiscal year (either as continuing, new, or disaster), a record is generated and is never removed from this file. This record contains the main component of the application and payment file, consisting of receipt control, accounts receivable, and ID information. Even if the applicant ultimately does not qualify for or receive payment, the file retains this main component. Depending on what section(s) of the law the applicant applies under, the application will acquire Table 1 through Table 10 data (including, possibly, properties claimed) (see *Attachment H*). Then summaries are computed, creating the TR21 or TR22 data. Finally, if payments are appropriate, payment data are computed, using current annual figures and application data (TR21, TR22).

The problem arises when changes are made to one or more of these data fields after the LEA has received payment based on fields that are now known to be erroneous, but before the books are permanently closed for that application. Changes are initiated in two ways--the data correction form (TR21, TR22), or the field report, which changes either the same data as the TR21 or property claimed. The assumption made in this situation is that when such changes are made, all components of that application which are related to the changed field(s) will be updated accordingly. This means that these three transactions (TR21 and its derivations, field report update, and property 2 update

and correction) must be integrated so that the property update is performed at the same time as the field report update, and, if necessary, also performed along with TR21 updates. This will require the development of special-purpose programs to ensure consistency and aid the user in locating points in the data that are not consistent during transaction processing. An example of the behavior of such a program is provided in *Attachment J, Interactive Data Corrections.* In addition, when the change is actually made, the program will first write out all the pertinent data to the application and payment history file and will then update the current records.

From Batch to Interactive Transactions

Several transactions currently performed in batch lend themselves to being performed with a far greater degree of user interaction. These include:

- o having the property analysts update the Property 1 file directly and interactively rather than working through ADP, thereby eliminating all associated input forms entirely and the time delay associated with these batch requests;
- o eliminating the ID transaction (TR12) and the input form, to be performed by PSB interactively;
- o eliminating the following as batch transactions--Initial payment processing, Initial allotment, Amend allotment, and Percent update. These are all performed only occasionally and are quite simple; and
- o making the Cancel payment transaction obsolete because of new procedures instituted for controlling batch processing (discussed below).

The danger associated with automating transactions that were previously done as batch transactions is that data checking activities may suffer. To ensure data integrity, therefore, simple internal checks within OIA will coincide with the addition of this interactive processing of certain transactions. For example, two persons could check each other's work, the computer system itself could automatically generate on-line logs of changes done for an historical record, and simple programmed automated user-interfaces, very similar to the example shown in *Attachment J*, could be added. These records of

changes may only be necessary for a few days or weeks, depending on the particular situation. Finally, electronic input forms, capable of extensive user-defined edit checks, will also be used, just as they were with batch input processing.

A new system must take into consideration the inherent complexity of the current system, which consists of six interrelated systems--one for each of the past six years. Currently, there is a computer program (specifically, the payment transactions) for each year. If computations are to be performed, the program (or program module) containing the code for the relevant year must be selected. Such an organization could be implemented with the proposed system--it is a matter of integrating the program pieces so that the part specifying changeable parameters is selected correctly and the computations are performed utilizing the correct year's formulas.

The proposed system, however, would maintain the *changeable* parameters (items that differ from year to year) in a separate file, which could be manipulated by hand (editing) and read by the computer program which performs the calculations. This would reduce the need for re-programming every year, make the data more accessible and the entire system easier to understand. Depending upon the extent of the changes on the parameters each year, the files might be raw data files, macro-type program modules, subroutines which can be called by the main processing program, or a combination of the above. (This proposed plan utilizes the new file referred to as the Annual Control Data.)

Controlling Batch Processing With the New Data Base

In the current system configuration, batch processing is closely monitored and its pace is controlled by the payments section of the PSB, so that records such as batches sent out, number of applicants, and dates of request are maintained on a paper file (the daily control log). The majority of these batches fall into three types: 1) initial application processing (3,000 per year), 2) field report entries with possible data

corrections (2,000 per year), and 3) payment batches (TR21, TR22, TR28), which are performed at the rate of about 2 per week, 250 payments per batch, throughout the year (8,000 total entries). As mentioned before, automating these batch processes would necessitate new system components to maintain proper record-keeping, accurate processing, and accounting. To accomplish this, a set of interfaces would be developed that would allow OIA to cross-check the data, maintain information on the batches processed, and produce reports to be automatically printed when the batches are sent through. Thus, no matter who keys in the data originally, the OIA could determine its final entry into the data base in the manner and timing desired. In addition, this control would allow all the reports that are currently produced with batch transactions to be continued, and therefore maintain the same level of data checking. At the time that the payment transactions (TR21, TR22, TR28) are executed, the user (OIA) would directly enter the few fields now contained in the current batch header transaction sheet (hash total, date), and the batch transaction file would be updated, as is now the case.

In the cases of initial application processing or file correction via the field report, batch-type processing (large groups of cases combined) is not necessary. On the other hand, payment processing must be retained as a basic batch process because of the disbursement of money and transmission, via obligation tapes and payment disk packs, to the FMS. For either field report entries or initial application processing, if it is easier to perform these individually as they come in, this may be done; if waiting a few days and creating a "batch" is more convenient, this would still be faster than the current system, since it may no longer be necessary to go through a contractor.

Field report processing may be broken into two parts as indicated in the new file descriptions. The first part will consist of entering the field report log and the data corrections, and generating the data fields used to update the application and payment file(s); the second part will be the actual update. To the degree that applications are held up pending a field report, this update transaction may be entered individually as

the field report data are generated. Alternatively, waiting until several are ready may prove more expeditious. In either case, an automated process will be developed that will allow the OIA to perform the update when desired. This program will also perform the consistency checking necessary, write out the current record to the application history file, and update the current record(s).

In cases where large-scale data entry is needed, such as keying the applications (particularly new ones), the OIA could continue to contract out these services. Under the new system, an input shell file would be used and an input file produced, which the OIA could then enter via a pre-designed batch process. At this juncture, any reports needed would be automatically generated, just as they are now. The use of the input shell file, as described in this report, would greatly reduce the time, effort, and cost of keypunching the applications, regardless of who does the actual work.

There are several queues in the application and payment files at all times (within one fiscal year), reflecting the various states that application may be in at any given time. These queues consist of: applications which have not yet been completely verified, applications with complete data waiting for the year's first allocation, LEAs for whom an obligation has been established but which cannot yet be paid (estimated payments), and LEAs who are cleared for payment but have not yet received it. In addition, there is a queue of LEAs with other problems, which are held pending further data verification or a field report. (See *Attachment K, Schematic of Batch Processes.*) The first process takes place before the annual allocations; once the dollars are allocated, obligations may be established immediately or payments made, with the initial payment transaction. Establishing this obligation is important for the accounting system; these figures are sent to the FMS as dollars claimed on the obligation tape. Most LEAs are immediately eligible for payment; however, for a few, an obligation is established but no payment is yet warranted. For those ready for payment, the computer must then select groups which are geographically distributed.

In the current system configuration, the distinction between calculating an estimate and calculating a payment is achieved via the payment code on the TR21 transaction form. In addition, the TR21 form is used to make corrections to data already entered via the application in the final visual screening. Therefore, TR21 forms serve four purposes: 1) to keep applications with obviously erroneous data from receiving payment; 2) to hold applications which probably have good data but which must be held for other reasons (the estimated payment); 3) to correct (update) data from either of the above cases; and 4) to generate actual payments and payment types, either with or without performing an update. Combining these separate functions would allow corrections to an application and generation of a payment immediately upon making the correction. However, SAB must still have the ability to perform a final visual check on all payments generated before the payment disk pack is produced.

To the extent that having the TR21 forms in hand is helpful in spotting errors, they should be retained. However, in the new system, temporary files would be used which contain TR21 data and payment calculations for visual inspection and editing, and which will mirror exactly what the paper forms contain (the TR21 transaction file, the follow-up payments file, and the preliminary applicant file). Therefore, the complete functional equivalent of the paper forms, as well as a replacement for the old "Stop Payments" transaction, will exist. This latter process could replace the paper copies entirely if desired, but does not have to. If paper copies are desired, then TR21 transactions, both to change data and to generate payments, can be directly performed on-line by OIA personnel using the paper forms. This would still be considerably faster than using contractors to perform the tasks.

Transactions corresponding to the above situations that must be retained are: initial payments, follow-up payments, corrections from the TR21 (TR22) form, and field report corrections. These transactions would be achieved in the proposed system by means of three files, all of which would be temporary and retained only until the main

file update is performed. (See *Attachment K, Schematic of Batch Processes.*) These files will allow the functions which are now done via the paper forms to be performed on-line, if desired, by editing these files. Of course, once initial payments processing occurs, the current TR21 data will remain in the application and payment files for checking whenever desired.

Two other critical transactions, the 5-card from FOB and the TR2-card from property, should be retained for cross-checking application accuracy. These could be performed instantly and interactively by these two sections of the OIA, from their own terminals, requiring the development of a small and simple user interface. The overall application processing would thus be further speeded up. For a complete inventory of transactions needed with the data base, see *Attachment L, Description of Transactions With New OIA System.*

New Essential Features

Several features will be essential in a new system that is designed to respond to identified problems with the current system. These features include:

- The ability of various persons in OIA to directly request predesigned reports (any of those listed in *Attachment E, Computer-Generated Reports*) through terminals, and get them quickly (10 to 15 minutes). This would involve the creation of an automated menu of all possible reports.

- The ability of the SAB to query the system and get *quick* (one minute or less) answers to common questions, such as the status of an application from the current year in the receipt control process, how much money it will receive if a payment has been calculated, the last time an LEA received Impact Aid, and when an LEA last had a field report.

- The ability to use the pre-printed application as a shell for the new data when it comes in from the LEA so that only new or corrected information need to be keypunched, rather than having the entire application punched over again. This means the creation of an automated input form, based on the pre-printed applications for that fiscal year.

- The addition of receipt control and history (application) information for disaster and construction payments, so that these two subsections can share data with Sections 2 and 3,

and perform many of the functions now restricted to Sections 2 and 3. In particular, the disaster computations could be performed on-line, cases could be kept open and pending with their own computer file, and when final costs are entered later, adjustments to payments could be made on-line.

- The addition of the Common Accounting Number (CAN), now used only for Sections 2 and 3, to obligations and payments for data generated for disaster and construction to aid the FMS in accounting for dollars spent by OIA. All money appropriated, obligated, and spent will be kept by CAN in this manner in the control ledger files.

- The ability to perform ad-hoc analyses and reports on various parts of the data base whenever desired (statistical tables, aggregations, merges of different parts of the data base), and print them in the OIA.

- The use of an on-line record of requests for new applications that come into OIA throughout the year. This record can serve two purposes in addition to permanent documentation: (1) new applications can be generated more easily at the start of every fiscal year because some pertinent data will already be in the system, and (2) when new applications do come in, this on-line log can be used to feed into the field report decision-making process done by SAB and required for all new applicants.

- The use of DOS-based microcomputers as part of the system, so that other office automation systems developed in the future can interact with the OIA system and so that small subsets of data can be further manipulated with PC software (i.e., spreadsheet analysis, word processing, or graphics). This includes the ability to upload and download data to and from the main system and PCs.

- A design of sufficient flexibility so that new files can be added in later years and incorporated into the data base, such as new types of construction assistance requiring different data, or a change in the data needed for eligibility calculation.

- Increased interaction with the data contained in the FMS MIDAS system so that feedback on appropriations, obligations, and payments can be provided to OIA, thus ensuring that the two files agree. Since the FMS is the authority on budgetary allocations, they must provide OIA with regular updates on the current status, so that OIA can reconcile discrepancies. Either the OIA must be given read access to that part of the MIDAS data base, or FMS must provide OIA with regular reports or data sets containing information needed to do this.

- The elimination of the paper receipt control log kept by the FOB, in favor of direct access and updating of the on-line receipt control log, which is part of the application file every year.

- The elimination of the paper TR21 submission process for pre-applications done every year. There already exists, on-line, a file with data on all applications receiving aid the previous year (this is used to generate the pre-printed applications originally). If these applicants submit a proper letter, if they received money the previous year, and if a pre-printed application has been sent to them, then the SAB can pass them on for immediate approval for payment.

- The maintenance of the field report log, which is a record of all field reports, on-line rather than on paper. This information can then be transferred to the receipt control file directly and used for the field report update transaction.

- The development of a simple input form which is usable on microcomputers (using dBASE IV, Lotus, or the like) that LEAs can use to input the survey forms data (which all Section 3 recipients are required to submit annually). A small program can be added to this input form which sums up the child counts into the proper categories; this could be done either in the field by the LEAs or at OIA. Floppy diskettes could be directly mailed to OIA. Another small input program could be developed enabling this data to be directly entered into the receipt control portion of the main data base and used for field report updates if appropriate.

- The ability of the SAB to edit and update the annual parameters directly on-line. This includes the appropriation amounts, percent entitlements, and prorating percentages as well as current edit checks in force. Security checks would have to be incorporated, as well as built-in guards against accidentally changing critical data. These annual figures would be stored in the annual constants file, listed in *Attachment F, Logical Data Groups With Keys*. Six versions of this file will be kept, corresponding to the last six years of changes in the law and the computation of payments.

- The addition of a validation process whereby EIN numbers are checked, just as IDs now are, upon entry of applications. This will require that OIA obtain read access to files in the MIDAS system containing those identifiers. This will ensure consistency between regular Sections 2 and 3, disaster, and construction applicants.

New Desirable Features

In addition to the features described above, other capabilities are desirable but not essential. These include:

- The ability of field officers to enter data from local sites when doing field reports, and send it to OIA for immediate checking; the ability to interact with those in the field and provide them with current data pertaining to the LEAs.
- The ability to develop formulas for computing the most efficient scheduling for field office visits, based on factors such as time since review, dollars spent, geographic location, and proximity to other LEAs.
- The ability of the LEAs to enter their own application data and do some of their own data checking on local microcomputers, such as checking the legality of properties based on files provided by OIA.
- The ability of LEAs to send their application data directly to OIA, through the phone lines or mailed-in floppies, eliminating the need for the paper form.

HARDWARE/SOFTWARE/PERSONNEL REQUIREMENTS

The above description of the full range of required system capabilities points to a set of hardware, software, and personnel requirements. These requirements are discussed generally below as they relate to the system requirements. Part III, System Alternatives, will present specific hardware and software recommendations for the proposed alternative systems.

Hardware/Software

Given the system requirements, a data base management system (DBMS) is recommended. No other type of software combines the features of quick data retrieval, non-programmer interface, great flexibility for re-organization, growth of the data base, and ability to store the large-scale files used by OIA (50 to 100 megabytes, perhaps larger). In addition, many DBMS systems allow changes and updates, in batch mode or interactively, and access through traditional programming languages (COBOL, PL/1,

FORTRAN) for more elaborate interfaces. This type of flexibility is not matched with any other type of software.

Within the framework of a data base management system, more specific requirements for a new system include:

- o structured Query Language (SQL) or an equivalent fourth-generation type language for easy, ad-hoc user interface by non-programmers;
- o support for a third generation programming language capable of full data base manipulation (probably COBOL);
- o uploading and downloading to and from DOS-based PCs;
- o support for up to 20 simultaneous users;
- o storage for files of the size and structure listed above;
- o logs of some transactions performed, including interactive ones;
- o a reasonable turnaround speed for requests, reports, and other processes;
- o support for local printers so that reports can be directly obtained by OIA in the office;
- o security features such as various users having system IDs, and, based on these, being only allowed access to specified portions of the data base, by both user-defined views and different functions allowed. The ability to designate read-only or read-write access;
- o the ability to perform standard data base functions beyond search, add, delete, update. These include sort, merge (joins) by one or more variables, and, in a variety of ways, aggregate numeric variables and save the files created;
- o backup and security features to protect against disk error, system failure, and other hardware/software problems;
- o the ability to design complex interactive input screens that can do extensive user-defined error, consistency, and ID checking;
- o the full range of mathematical operations on whole numbers with dollars and cents, and with packed decimal arithmetic; and
- o the ability to output character-type files as needed for downloading to PCs.

Personnel

A new system accessible by OIA staff will require the assistance of a person with specialized skills to administer the system. This person would be the designated data base administrator, through which many requests would be channeled. This person's duties would include: playing a major role in the physical design and implementation of the data base; programming; contracting keypunching when necessary; helping design reports, input forms, and other documents; taking requests from other OIA personnel and carrying them out; setting up security and access standards and rules for the data base; ensuring that the data base is backed-up regularly; and training other non-technical OIA personnel in the use of the data base. In practice, this person could be in OIA, in OIRM, or available through a contractor.

ISSUES

Finally, examining system requirements raises many issues, some of which have been previously discussed and others of which will be addressed in Part IV, Implementation Plan. Key issues include:

- As OIA personnel assume more control and direct access to the data base, how will the extensive checking for accuracy now done by numerous persons be retained? Changes should not become so easy to perform that proper cross-checking by other persons or re-checking by the same persons is sacrificed. Personnel guidelines, as well as proper computer program design, will be required to insure the integrity of the data base.

- To what extent should paper records and documents be kept available, even if the level of automation of the system is increased? For some purposes, there is no substitute for paper records (it provides legal documentation, often provides good back-up records, and others without access to computers can use it).

- In which instances is it safe to change data permanently vs. keeping a history of all that has taken place, particularly on-line? What should be the relationships between such history data and paper records that are kept?

- To what extent should the existing batch-oriented system be retained vs. implementing a more interactive, time-sharing type of system? For some of OIA's work, transactions are highly structured and do not change (e.g., entering new applications) and batch processing may have advantages in several of these instances. Some of these advantages include better self-documentation; preparation of jobs independently of processing them; re-using programs and commands more easily; less likelihood of job interruption and, therefore, error; and increased machine efficiency.

- Any time a new system is implemented, questions of consistency with past systems are raised--what is the proper balance between consistency with past schemes vs. ideally designed systems? For example, FIPS codes to represent states are used by most data processing systems, but changing them will cause some adjustment, inconsistencies with past data, and extra programming.

- How much should the ability to implement the new system incrementally determine its design?

PART III - SYSTEM ALTERNATIVES

OVERVIEW

In determining the range of possible alternatives for the OIA computer system, several key factors must be considered. These were described in detail in Part II (System Requirements) and are summarized below:

- o use of a data base management system (DBMS),
- o capacity of over 100 MB of file storage (including estimates for growth),
- o capacity for 20 simultaneous users and approximately 50 users overall,
- o quick accessibility of large portions of the proposed data base (perhaps one-third to one-half), and
- o frequent accessibility to the data base (several times a day).

Taken as a whole, then, these factors point to the requisite hardware environment in which the OIA can operate. In general, a consideration of hardware environments includes microcomputers, minicomputers, and mainframe computers, as well as combinations of these three categories. The conclusions of this feasibility study are that:

- o A microcomputer system alone is not feasible given the requirements of the OIA system, summarized above.
- o A minicomputer system by itself would be possible if it had a large capacity and was dedicated to OIA. Although there are some minicomputers available to OIA--Wang VS100, Data General MV1000, and Prime Super-Mini, all are medium-scale computers and none are entirely dedicated to OIA. Investigations of the available minicomputers revealed that either the machine is no longer supported by the manufacturer, does not support full-scale data base management system software, is too small in capacity (for either current system functions or for future expansions), or the OIA does not have exclusive access.

  Therefore, the use of minicomputers was excluded from further consideration in an alternative system.
- o A mainframe computer is required for any proposed OIA computer system. The ED mainframe computer facility is currently located at Boeing Computer System (BCS) in Vienna, Virginia. This mainframe system:

- -- is an OS/MVS IBM Model 3084/QX, supporting TSO and NIH Wylbur,
- -- has virtually unlimited disk storage,
- -- has tape drives compatible with 9-track and the newer 3800 cartridge tapes,
- -- supports local high-speed printing, both dot matrix and XEROX 9700 laser quality (printing can be achieved via the dedicated high-speed printer in ED, supplemented by 2400 baud dial-up printers, or through dial-up printing alone),
- -- provides 1200 and 2400 baud dial-up asynchronous ports and front-end processors for higher speed synchronous, dedicated lines (such as those for remote job entry (RJE) systems), and
- -- supports three widely-used data base management systems--Model 204, IDMS/R, and Focus.

Finally, in considering a mainframe system and accompanying DBMS software, it should be noted that should ED rebid its mainframe computing services in the future, any DBMS available at Boeing would most likely be available at or transportable to any other standard IBM or IBM-compatible facility.

- o In addition to mainframe capabilities, the inclusion of some degree of local microcomputer support is also desirable for OIA's computer system. Several factors indicate the desirability of local microcomputer support, including
  - -- Boeing's response time may not be fast enough for OIA needs during the busiest hours of the day (11:00 a.m. to 5:00 p.m. approximately) because there are hundreds of users on it,
  - -- any mainframe system goes down occasionally,
  - -- some of the simpler processes that OIA performs do not require a mainframe, and
  - -- other office automation systems that OIA may construct in the future could be used in conjunction with local data base processing.

The above considerations can be accommodated by two types of system configurations, both of which would satisfy the needs of the OIA, and make maximum use of available resources. These two systems are:

- o a mainframe-based DBMS, accessed both through local terminals and DOS-compatible PCs, with the microcomputers as important adjuncts for local data input, small-scale simulation, report generation, and interface with other office systems, and
- o a combination mainframe/microcomputer system with a local area network, in which data storage and processing are shared between the mainframe and DOS-based microcomputers, and locally stored data and software are shared, via local area network (LAN) software, between the various microcomputers in OIA.

The following sections provide a detailed description of each of these two alternative systems, followed by a comparison between the two systems. Four attachments in the Appendix provide further clarification: *Attachment M, Schematic Design of the Mainframe-Only System; Attachment N, Basic Components of LANS; Attachment O, Illustration of Software for the Micro Mainframe Combination System;* and *Attachment P, File Implementation in a Mainframe/Microcomputer System.*

ALTERNATIVE A: A MAINFRAME-BASED DATA BASE MANAGEMENT SYSTEM

One alternative system for OIA would be a mainframe-based system, with PCs as terminals in the local office. Some PCs would be used primarily as terminals, while others would be provided with capacities to perform some other processing, input, and analysis functions. This configuration would have the mainframe system installed at Boeing, and would include approximately 30 DOS-based PCs and two local printers. The processing, DBMS software, and data itself would nearly all reside on the mainframe. The PCs would be used for processing small files after downloading and for entering data that are to be uploaded and run against the main data base on the

mainframe. Interactions would take place by logging on to the mainframe system, via the current LAN in OIRM, and invoking the DBMS software from there.

Components of a Mainframe-Based System

The mainframe-based system is by nature far simpler in its configuration than the combination mainframe/microcomputer system. The components required are:

- the remote system itself, where processors, disk storage, main memory, tape drives, and I/O systems reside;
- local hardware, including 30 connecting points to the existing LAN and necessary hookup cables;
- one LAN card to be installed in each PC; and
- a set of IBM-compatible microcomputers, with DOS operating system, word processing and basic utilities, and optional additional software, such as Lotus 1-2-3, dBASE IV, or similar software.

In addition, two local printers are recommended--both laser printers to be attached to one or more of the microcomputers. Both these printers would be used to supplement the high-speed printer in ED. (See *Attachment M, Schematic Diagram of the Mainframe-Only System.*)

The proposed mainframe system would follow the pattern of most traditional DBMSs:

- The data base files, including actual data as well as the data definition files, are all completely integrated in one system,
- Batch transactions are performed by preparing a normal jobstream, either as a set of commands in the DBMS query language or in a traditional high-level language which interfaces with the data base query language,
- Input forms, reports, and data base views are pre-defined and standardized from the point of view of the user,
- Interactive input screens and menus can be written in the high-level language to provide the easy-to-use interface needed for most users, and
- Direct queries are easy to perform, using the query language of the DBMS.

Such a system is capable of creating its own complete environment on the mainframe because the software offers the set of tools needed for all normal functions. Like most mainframe systems, the Boeing system has several other software tools which can be used in conjunction with the data base if desired; usually some special-purpose manipulation or programming is required for the interface. The system proposed here is a slight enhancement of the traditional model in its use of microcomputers for some simple functions, thereby removing some of the demand on the mainframe, increasing OIA's flexibility, and allowing future changes and expansions.

For individuals doing direct query or updates to the data base, or in any other way accessing the data base directly, the terminals would suffice. However, flexibility is enhanced through the availability of microcomputers to perform data entry with some checking (using dBASE), preparation of programs (using editors), and small scale manipulations of output data (such as the payment simulations now performed in Lotus or the generation of Congressional letters). In addition, the microcomputers allow a degree of flexibility with any other future office automation systems that may be instituted in OIA. Further, should a new generation of microcomputers and/or software become appropriate as a means to implement the data base in the future, the OIA would already possess some of the equipment and training for such a system.

Mainframe DBMS Software

Currently, there are four DBMS's available at the Boeing facility--Focus, IDMS/R, Model 204, and System 2000. System 2000 was eliminated from consideration immediately because the technical personnel at the Boeing facility reported that the system is an old version and is no longer supported by the vendor.

It is noteworthy that the Boeing system does not support any true relational DBMS; the software types supported there all fall into the category called the "network model." However, this is not a serious drawback for the application at hand because

most of OIA's functions require the very kinds of operations for which the network DBMS is best--speedy retrieval and update, using pre-defined paths of access into the data base. Operations that require relational power can also be performed, only somewhat more slowly.

Given the DBMS software available at Boeing, Model 204 is recommended (while acknowledging that IDMS/R could quite possibly perform the needed tasks as well.) Focus was eliminated because it is notoriously slow in responding and it does not have a well-developed interface with high-level languages, a basic software requirement for a new OIA system. Model 204 is recommended over IDMS/R because it:

- has an interface with microcomputers so if more computing on microcomputers were desired in the future, the transition would be easier than if one had to program the interface without this feature,
- has a very high level of support at Boeing,
- is widely used by other government departments, including ED,
- is currently available to ED on the Boeing comp. er facility, requiring no further contract modification, and
- fulfills all the listed software requirements.

Summary

To summarize Alternative A, the mainframe-based system, the recommended system would consist of 30 DOS-based PC workstations, 10 having some processing capacity and 20 functioning as terminals only. All would be cabled to connecting points on the existing ED LAN for accessing the mainframe. The system would include two locally connected laser printers that could be shared among the PCs. Other software available on the PCs would include WordPerfect, Lotus 1-2-3, and dBASE IV. The software recommended for this mainframe-based system is the Model 204 data base management system. Local hardware is as follows:

- o PC workstations (10)
  - AST Premium 286 PC or other IBM clone with:
    - 1 Mb RAM
    - 40 Mb hard disk drive
    - 1 Hi-density floppy drive
    - Monochrome monitor
    - MS or PC/DOS operating system
- o Simple PC workstations (20)
  - Any of several on the market with 2400 baud capability
- o Local printer (2)
  - HP Laser Series II with:
    - basic 8-font cartridge
- o Optional software for the PC workstations
  - WordPerfect
  - Lotus 1-2-3
  - dBASE IV
  - Harvard Presentation Graphics

ALTERNATIVE B: A MAINFRAME/MICROCOMPUTER SYSTEM WITH A LOCAL AREA NETWORK

A second alternative to answer OIA's needs is a computer system which uses a combination of Boeing's mainframe processing and storage, microcomputers for further processing, and a local area network (LAN) so that the microcomputers can share data and software. The files needing quick access, i.e., application status, payment records, and field report information, would be stored on the microcomputer, while files having large storage requirements or needing a significant amount of processing time would be relegated to the mainframe environment.

Although the microcomputer has many outstanding features, it is limited in two ways:

- o it is normally a single user system, which makes the sharing of information difficult, and
- o its capacity and power are quite small when compared to minicomputers or mainframes.

Recent advances in microcomputer technology have sought to overcome the first problem through the introduction of local area network architecture, which allows a series of PCs

to be linked together electronically. The LAN system permits multiple user access and the ability to share files, software, and other peripheral devices such as tape drives and printers. In addition, specialized PCs can be placed on the network to perform specific functions, leaving individual PCs free for other tasks. Further, the power and size limitations are slowly being overcome as technology advances.

Components of a Local Area Network

The basic components of LANs (shown on *Attachment N* in the Appendix) are: (1) PCs functioning as network servers, (2) PCs as user workstations, (3) interface cards, and (4) connecting cables. Network servers are the foundation on which most networks are built, and they are critical because they determine the speed, security, and convenience of the entire network. The network server is a PC dedicated to running the network operating system, maintaining user directories, providing system security, and storing files that are to be accessed across the network. The network server should have a high-speed processor and a hard disk capacity that allows for file storage of the system and room for expansion. Having a network server that is too small will seriously degrade the performance of the network.

A server is a centrally located microcomputer with one or more workstations attached. They are special machines which exist primarily to service the requests of the other PCs on the system and make their attached disk drives, printers, modems, and other resources available to the individual workstations. The networking software resident on the server determines whether the server will be dedicated to its service role or whether it can also run application programs. Network servers are usually 80286 or 80386 machines, these being the newer and more powerful generation of PCs. The major companies that provide the networking software, Novell and 3COM, increasingly feature dedicated servers with 80386 processors.

Workstations on the network are individual PCs that are linked to the network server by the interface cards and connecting cables. The workstation can access all shared devices, both hardware and software on the network, and it can perform processing on its own processor independently of the network.

The next component of the LAN hardware is the interface card. Within each server and workstation, a network interface card is installed which permits communication with the networking software through special driver software. Size and on-board processing power of the network interface cards can vary greatly. The major interface cards include Ethernet, Token-Ring, and ARCnet.

The final component of the network is the connecting cable. The interface card determines the type of cabling needed to connect the servers and workstations. Choices include twisted-pair telephone wire, shielded twisted-pair telephone wire, coaxial cable, and fiber optic cable.

Recommended LAN Hardware and Software for the Mainframe/Microcomputer System

The evaluation of a local area network approach to address OIA's needs to access data and retrieve information will focus on currently available LAN hardware and software. The components of the LAN can be configured in a variety of ways to form an optimally productive network, and there are a number of wiring and hardware schemes available for implementing a LAN using IBM or IBM-compatible machines. In an environment where the number of potential workstations on the network is small (30 or less), a network based on the Ethernet standard is recommended. Ethernet has the following advantages:

- Speed - Ethernet is designed to transfer data at the rate of 10 megabits-per-second, the fastest data transfer currently available.
- Availability - There are a number of manufacturers who make the necessary hardware for using PCs on an Ethernet-based network, making it more available and less costly than others.

- o Proven Track Record - Ethernet is one of the original networking standards with endorsements from Digital Equipment Company (DEC) and Xerox.

- o Compatibility - Ethernet allows the interconnection of a number of varied computers such as IBM, Apple, MacIntosh, and DEC.

The major network operating systems currently available for use with Ethernet include Netware from Novell, 3Plus from 3COM Corporation, and PROFS from Ungermann-Bass. Since a LAN currently exists in ED using the Ungermann-Bass software and broad-band dual cabling, the proposed network would be implemented as a supported sub-LAN, residing under the current LAN. In this way, the Ethernet standard could be used if desired. The components of the current LAN to be used by this proposed alternative system include the outside access ports to the Boeing system and most of the local cabling now in existence at ED. The remainder of the design of the proposed LAN for OIA is independent of the current LAN. *From this point on, references to LAN refer to the proposed OIA sub-LAN, unless stated otherwise.*

There are two basic approaches for implementing a network to meet OIA's requirements. In the first approach, the data base is maintained on the network server and every workstation on the LAN has access to the data base. This approach closely resembles a multi-user data base on the mainframe; however, the limitations of this approach include: (1) data base management software developed specifically for the LAN must be used; (2) programming such a data base application can be complicated by the security requirements placed on the data base files; and (3) all input/output requests must be made over the network, which can degrade the performance of the data base by overloading it.

The recommended approach makes use of a second dedicated machine to act as a data base server, in addition to the PC which acts as the network server. This second machine would maintain the data base on its own hard disk and respond to requests made by the workstations. The advantages of this approach include: (1) LAN specific

data base management software does not have to be used; (2) the data base can still be available even if the network is out-of-service; and (3) this setup provides for future expansion through the use of other dedicated servers.

In addition to the network and data base servers, a remote access terminal is recommended, with its own dial-up modem. This terminal's purpose is to respond to requests for information from the field or input a field report directly into the system from the field. Field Office personnel would require access to a PC with a modem, from which they could call in to this machine. This type of equipment is becoming more and more common in local scho l districts.

Workstations can be any IBM or IBM-compatible machines. The use of 286 machines is recommended because of their speed and their reasonable price.

Mainframe/Microcomputer Data Base Software

The major criteria for determining the appropriate data base management system software for the mainframe/microcomputer LAN configuration include:

- the availability of similar or compatible data base management software for both environments,
- the efficiency of transmitting data to the mainframe from the microcomputer and vice-versa,
- the relative ease in retrieving data from the system by personnel that may not be computer literate, and
- the ability to develop the necessary user input screens.

While there are many DBMS products for the microcomputer, choices are limited by the availability of compatible data base packages on the mainframe. This discussion will center mainly on microcomputer-based data base management systems since a discussion of the various mainframe data base management systems available for use with the OIA system were detailed in Alternative A. Of those mainframe DBMSs available, Focus and System 2000 were eliminated, leaving IDMS/R or Model 204.

Model 204 has available a PC version known as PC/204. The PC version is not a "stand alone" data base management system; however, it can be used for microcomputer/ mainframe communications. For example, if Model 204 were the chosen DBMS for the mainframe environment, PC/204 could be used to retrieve selected data from the mainframe data base and download it to the microcomputer for loading into a microcomputer DBMS such as dBASE IV or Rbase System V. Conversely, PC/204 will act as a conduit for uploading data from the microcomputer to the mainframe. This is an important aspect of the new system since data updates and corrections will be entered into menu driven input screens developed for the microcomputer and then uploaded to the mainframe.

The other candidate software is IDMS/R on the mainframe and Oracle on the microcomputer. This software configuration is not compatible; data cannot be uploaded or downloaded directly into or out of the mainframe DBMS. However, interface programs can be written in Oracle that permit data to be easily transmitted from one environment to another, thereby making the mainframe system and the interface invisible to the user.

Both software systems could adequately perform all functions required to implement the OIA system, and neither software package has a significant advantage over the other. (See *Attachment O. Illustration of Software for the Micro-Mainframe Combination System.*) However, the Model 204 system on the mainframe is recommended, for the reasons cited under Alternative A and because of the added advantage of the PC/204 software for selecting and transporting data to and from the microcomputer. In addition, the dBASE IV data base management system is recommended over Rbase System V because dBASE IV is more widely known and is already used in ED.

Summary

To summarize Alternative B, the recommended configuration for the mainframe/microcomputer local area network (LAN) combination is illustrated in *Attachment N.* The LAN will be based on the Ethernet standard and will function as a sub-LAN running under the existing system. The hardware components consist of a network server, a data base server, a remote access terminal, a laser printer, and workstations for the various branches of OIA. The recommended data base management software for the mainframe will be Model 204 and will use PC/204 for microcomputer to mainframe communication; the microcomputer DBMS will be dBASE IV. In addition, a modem and telecommunication software will be available on the data base server for connectivity to the Boeing Computer System. The distribution of the recommended hardware and software is as follows:

- Data Base Server -
  Compaq 386 20MHz PC with:
  2 Mb RAM
  140 Mb Hard disk drive
  1 Hi-density floppy drive
  Monochrome monitor
  3COM Etherlink board
  dBASE IV data base management system
  PC/204
  WordPerfect
  Lotus 1-2-3

- Network Server -
  AST Premium 386 PC with:
  2 Mb RAM
  70 Mb Hard disk drive
  1 Hi-density floppy drive
  Monochrome monitor
  3COM Etherlink Plus board

- Remote Access Terminal
  AST Premium 286 PC with:
  1 Mb RAM
  1 Hi-density floppy drive
  3COM Etherlink board
  Everex 2400 asynchronous modem w/dial-in modem

- Printer
  HP Laser Series II with:
  2 Mb Memory upgrade

- Workstations
  - AST Premium 286 PC or other IBM AT Clone with:
    - 1 Mb RAM
    - 40 Mb Hard disk drive
    - 1 Hi-density floppy drive
    - Monochrome monitor
    - 3COM Etherlink board

- Optional Software for the Workstations
  - WordPerfect
  - Lotus 1-2-3
  - dBASE IV
  - Harvard Presentation Graphics

In addition, the LAN will have the capability to communicate with the Boeing Computer System mainframe through the existing LAN's connection.

File Implementation in a Mainframe/Microcomputer System

In order to completely describe the mainframe/microcomputer system proposed here, it is necessary to describe the division of files and processes between the two environments. The proposed setup includes: (1) files on the microcomputer, (2) files on the mainframe, and (3) files or portions of files maintained on both, as well as a rigorous routine of uploading and downloading files regularly to maintain file consistency and integrity. The type of information that will be available for quick reference includes:

- the current status of an applicant;
- the current as well as prior years' payment records for each applicant, including any accounts receivable;
- the applicant's Section 2 and/or Section 3 summary data (TR21,TR22);
- state level summary data, i.e., payment records aggregated to state level;
- batch payment data, summarizing the payment processing throughout the year; and
- the field report log file, containing all field report information.

Attachment P, File Implementation in a Mainframe/Microcomputer System provides a discussion of each file and the environment in which it will reside. (For more details on each file, see *Attachments F and G.*)

COMPARISON OF THE TWO ALTERNATIVES

Two alternative systems have been discussed--a mainframe-based system and a mainframe/microcomputer system with a local area network. Each alternative has advantages and disadvantages in its ability to fulfill OIA's needs in a new computer system. The strengths and weaknesses of each system are presented below:

- Simplicity - Two areas where this criteria applies are hardware and software; in both cases, the mainframe system is simpler. First, there is less hardware to purchase and, therefore, lower initial cost. Second, less software is necessary because only one DBMS is needed for essential system operation. Other software use (such as dBASE IV or Lotus 1-2-3) would be the same with either system.

- Implementation - Designing all the details of either system will be a very labor-intensive process, with hundreds of hours of programmer and analyst time; this would be reduced considerably with the mainframe-based option. There are fewer programs to write, fewer interfaces to build, and only one rather than two DBMSs to learn.

- Maintenance - The mainframe-based system requires fewer routine maintenance procedures, if only because the regular upload-download required by the mainframe/microcomputer system would not be necessary.

- Data Base Integrity - The mainframe-based option is stronger on this criteria because whenever two copies of the same data are kept, there are potential problems of consistency. These can be largely overcome with careful planning and data base maintenance, but, nevertheless, the mainframe system does not have this problem.

- Security - Protection from unauthorized access, use, or destruction is always a consideration with program data of the type OIA keeps. There is somewhat more of a chance that security will be lost with data stored on microcomputers for two reasons: micros are easier to break into and usually more available, and two copies offer twice as many opportunities as one.

- Data Back-up - The Boeing system performs its own automatic weekly back-up of all disk files on the system, which would be in addition to that performed by data base administrator. This affords an extra measure of security for the mainframe-based system because with the mainframe/ microcomputer system, some of the data (the most recent updates) would not necessarily be on the mainframe at the time of the back-up.

- Technical Support - With the mainframe/microcomputer system, OIA would be called upon to provide their own hardware-software support for those components of the data base that are on the microcomputers. This support could be provided by a data base administrator. On the other hand, the mainframe-based system takes complete advantage of the technical support provided by Boeing for nearly all aspects of system maintenance (although this may be limited to business hours, and is also limited by how adequate that technical support is). A further advantage to the mainframe-based system is the fact that others in ED are also using Model 204.

- Experience - Mainframe systems very similar to the one proposed here have been implemented and used successfully for over 20 years; networks like the one proposed are at the cutting edge of new technology. While the mainframe/micro-based network model will undoubtedly become more standard over the coming years, there will inevitably be more complications with such a system--there simply has not yet been the time to resolve all problems. For current reliability, the edge goes to the mainframe-based system; for flexibility and the ability to adapt, grow, and convert into possible future systems, the edge goes to the mainframe/micro-based network.

- Hardware Maintenance - With the network, an elaborate system of cabling is called for within OIA offices. While this problem may seem trivial, the laying out and setting up of this cabling can be a problem. If it is not done very carefully and correctly, or if the cabling is ever disturbed, problems can ensue. For example, knocking a cable loose from a connection can cause the entire network to go down and data to be lost.

- Log-on IDs - It is possible that the Boeing system is unwilling to issue the log-on IDs necessary for OIA's needs; approximately 30 might be required. This is not a problem with the mainframe/microcomputer system because only the data base administrator would ever log directly onto the mainframe.

- Data Availability - If all the data are kept on the mainframe and it crashes, then no data are available at all until the system is backed-up and running. With the mainframe/

microcomputer system, however, much more work could be done in OIA, including most data queries and report generation. Although the mainframe-based system includes some work on microcomputers, such as preparation of input data, their use, if the mainframe goes down, is more limited.

- Speed of Access - It is likely that quick, easy access to program and application data would be accomplished somewhat better on the mainframe/microcomputer system for two reasons. First, with the mainframe-based system, unless one remains logged-on all day (this is usually not encouraged and does have cost implications), every time someone in OIA wants up-to-the-minute information, they would be obliged to log-on to the system; this takes perhaps one-half to one minute. Second, once logged-on, it is likely that queries and updates would be somewhat faster on the mainframe/ microcomputer system than on the mainframe, although this is difficult to predict and depends on other users on the system.

- Ease of Ad-Hoc Queries and Reports - As mentioned in Part II, System Requirements, the OIA needs the ability to occasionally request combinations of data, for either reports or analyses, that are not part of the ordinary transaction processing (for example, compare relative amounts obligated for the four sections of the law across the past five years). These kinds of requests are more easily performed with pure relational data bases, none of which are available on the Boeing system. Therefore, with the mainframe/ microcomputer system, it is likely that somewhat less work would be required for requests such as these, using software such as dBASE IV on the microcomputers.

- Training Personnel - While from a programmer's point of view, the mainframe is just as easy to use (especially since most interactions will be through specially-designed interfaces with menus), many non-programmers in OIA will need to use the data base from time to time. Experience shows that non-programmers are much more reluctant to use or try to use a mainframe, whereas many non-computer professionals find themselves drawn to and comfortable with microcomputers. With the mainframe/microcomputer system, non-computer professionals would never have to log-on to the mainframe.

- Role of the Data Base Administrator (DBA) - An important difference between the two suggested alternatives concerns the level of responsibility and, to some degree, expertise needed from the DBA. With the mainframe-based system, the DBA would need considerable knowledge of PC software and only a rudimentary knowledge of hardware. Hardware related tasks would consist primarily of connecting network points in the current PC hardware and, perhaps, purchasing PCs with the required configuration. Thus, the system could

be maintained satisfactorily with a fairly low level of hardware knowledge housed in OIA. The mainframe/ microcomputer system, on the other hand, would require considerably more skill and knowledge of general hardware facilities, particularly as they pertain to PC networks and interfaces. This system has many more PCs, complex local software (a second DBMS as well as network software), and extensive systems of cabling. Whether this is performed by OIA directly or OIRM, it will require a greater degree of hardware maintenance.

RECOMMENDATION

With the knowledge in hand at this time, Alternative A is recommended, while acknowledging that either of the proposed systems, if implemented correctly, would meet OIA's needs. This recommendation is predicated on the assumption that the following five criteria are of paramount importance. If, however, other factors are considered to be critical (such as the possible lack of responsiveness at Boeing), then the mainframe/ microcomputer system might be the preferred alternative. The five key criteria that favor the mainframe-based systrem are:

1) simplicity of the overall system (it is already complex enough),

2) greater ease of design, implementation, and maintenance,

3) security and integrity of the data base,

4) reliability, and

5) technical support available to OIA.

PART IV - IMPLEMENTATION PLAN

OVERVIEW

The implementation plan for an alternate computer system for OIA provides a set of specific guidelines for planning, designing, and implementing the new system, based on previous descriptions and analyses of the current system. The sections of this plan follow necessary chronology as much as possible. (*Attachment Q* in the Appendix is a time-flow diagram and displays the overall process. A three-year implementation period is assumed but a more rapid schedule could be used.) This document consists of the following sections: (1) preliminary planning; (2) structural specifications; (3) data conversion; (4) functional specifications; (5) maintenance; (6) development and testing; (7) documentation; and (8) phasing-in. The phasing-in of the new system would be planned as the last step so as to minimize the time required for both the old and new systems to be concurrently running; the entire system may be unavailable for perhaps one week.

PRELIMINARY PLANNING

Three key activities are required at the onset: (1) decide which proposed alternative system to implement and determine the accompanying software requirements; (2) finalize a position description for the data base administrator (DBA); and (3) specify the required hardware configuration.

Decide on Alternative System and Software Requirements

The two proposed systems are a mainframe-based data base management system (DBMS) or a mainframe/microcomputer system with a local area network. After it has been decided which system is to be implemented, then the programming language to be used for the main processing should be determined. The options include COBOL, PL/1, or possibly a different language such as Pascal or C if programming will be done on the microcomputers. COBOL may be heavily favored for this if it is decided that portions

of currently operating programs can be used in the new system. For software available at Boeing, the accompanying software documentation should be obtained from Boeing. For any additional software needed, particularly for the microcomputer, the software and its documentation should be acquired from the vendors.

At this point it would also be useful to solicit advice from persons in other agencies and/or offices in ED currently using the software, determine the person(s) at Boeing's technical support office most knowledgeable about the chosen software, and, in the case of the DBMS, establish support directly from the vendor as soon as possible (most large software companies provide this service).

Finalize a Position Description for the DBA

As discussed previously, the data base administrator fills a key role in the design and implementation of a new computer system. Therefore, the development of a position description for the DBA should be considered early in the planning process. Qualifications for the DBA should include:

- data base experience essential,
- experience in the DBMS software chosen very desirable,
- 2-4 years of programming in the high-level language(s) chosen,
- highly organized and detail-oriented,
- good oral communication skills,
- ability to work well with others, and
- interest or experience in the contents of the OIA data base or similar ones (that is, bookkeeping and administration of federal education programs).

Once the DBA has been hired, he/she can help with the rest of the planning process. Also, the roles of the staff in system development should be determined, particularly in the areas of the user interfaces and the conventions to be adopted. These lead persons should probably include division directors and individuals who fill out any

input forms or send in work requests for the current system. A committee should be established, composed of OIA personnel, the DBA, and representatives from OIRM, to help make decisions about the system as it is developed. In addition, some assistance from the current contractor will be essential.

Specify the Hardware

During the planning phase, decisions must be finalized regarding hardware specifications, including:

- the number and type of terminals and PCs,
- modems,
- phone lines,
- whether to use 2400-baud dial-up lines or a dedicated high-speed line and a local controller in the OIA,
- approximate amount of disk storage needed (on the mainframe),
- frequency of tape access (on the mainframe),
- number of simultaneous users, and
- number of sign-on IDs (on the mainframe).

Arrangements should then be made at Boeing so that the facilities will be available when needed. For example, OIA may require a dedicated disk pack for the data base. The staff at Boeing may also specify further technical details that OIA must consider.

STRUCTURAL SPECIFICATIONS

The key component of OIA's data system is the data base. Several decisions must be made regarding the structure of the data base, including descriptions of data elements from existing files, specifications of new files and data elements, various integrity constraints, and other structural aspects of the data base.

Descriptions of Data Elements from Existing Files

The final result of this step will be the complete description of each data element in the 29 specified components (files) of the new data base. Resources for this step can be found in *Attachments F and G* in the Appendix, and in the existing system documentation entitled the "SAFA User's Guide."

Most of the data fields in the new system will be taken from the existing files, but some fields will be eliminated. The total number of variables in all the files of the new data base will be approximately 2,000 to 2,500.

Once the new files and data fields are specified, the file sizes will need to be re-estimated, allowing for new fields in each data base file as well as data base growth. The estimates in *Attachment G, Calculation of File Sizes for the Impact Aid System*, can be used as a guide.

This exercise will result in a complete, ordered variable inventory for each of 29 files, including:

- o variable or field name,
- o type (internal representation),
- o length (where "type" does not completely determine this),
- o label or description of the variable, and
- o range or set of allowable values the variable can take.

In most instances, variable name and type can be taken directly from the old files, as many names are already descriptive and this makes the transition far easier. For new fields, particularly frequently used ones such as receipt control, new variable names will be required. Currently implemented edit checks, taken from programs which now process the data, can be used to start specifying allowable data value ranges.

In addition to describing the data elements, two general types of key fields must be specified:

- o those fields required to uniquely identify records (as listed in *Attachment F, Logical Data Groups with Keys*), and

- those fields which will be frequently used for look-ups and queries.

At this point any coding schemes that are to be changed, such as FIPS code or payment codes, should be determined.

For the latter type of key fields, the choice may depend on software used, which system is chosen, and the type and frequency of query. For example, it is likely that the field report log file will be queried not only by ID and FY, but by date within year or by region of the country. Two resources for this activity are *Attachment G. Calculation of File Sizes*, and the catalog of *Computer-Generated Reports* in *Attachment E.* Both of these provide guidelines for estimating the frequency and type zof access for the different files.

This process will result in a complete definition of the new data base in the data base definition language (DDL) of the software being used. Depending on the software and the preferences of the person(s) implementing the data dictionaries, it may be easier to specify the new data base directly in the DDL, particularly if the DDL allows for an easily legible listing of each variable name, type, length, and label. Then, the dual purpose would be served of documenting the data base, and defining and specifying it in the new software language.

Specifications of New Files and Data Elements

In addition to the existing files, there will be new data elements and files that will be features of the new system. Given the findings of this feasibility study, new components to be specified include (1) variables for the construction and disaster sections, and (2) an annual constants file.

For the construction and disaster sections, this step would entail creating an inventory of the variables needed for the construction and disaster sections and designing these new components. As noted in Part II, Systems Requirements, these new files will be subcomponents of the application and payment files, with one observation

per LEA per year maximum. Therefore, these entries will share IDs with the Section 2 and Section 3 files, and have the same main component containing the basic information about the LEA. These new files will be based on the data that are kept and used in those sections -- application information, receipt control information, and payment information.

The data elements in a file for the disaster section would, for example, include:

- some code identifying the federally-declared disaster, perhaps with additional information and description;
- receipt control information, with dates of events pertaining to the disaster and the application for assistance;
- information now contained on paper worksheets to itemize cost estimates;
- recommended payment, entitlement, insurance coverage, state contribution, and other fields pertaining to the payment calculations;
- identification as to whether an LEA is in a flood plain and is required to buy insurance;
- actual costs to replace items (these are currently obtained); and
- final disposition, payment, and any other information needed.

A decision must also be made on whether to enter previous disaster or construction data, or to begin entering only current information when the system is implemented. (If entering old data, additional input forms and keypunch instructions will have to be developed.) A reasonable compromise might be to automate only those cases that are still open at the time the system is implemented.

A further note regarding this phase of the system design is that any of the identified data elements and their definitions will likely change several times. As further stages of development reveal the need for more fields or different keys, for example, these changes can easily be accommodated. Similarly, if another type of Impact

Aid assistance were to come into existence in the future, files could be designed and added to the system at that time.

The need for an annual constants file was discussed in Part II of this document. This file would be a character-type file so that is can be edited and scanned interactively, and read by programs which compute payments. The file would contain three sections:

- o current annual parameters indicating allocations for each section of the law, with CAN number;
- o current edit checks in force to check for consistency, erroneous data codes, and other illegal values on the applications; and
- o current "percent entitlements" in force.

By dividing the file into three sections, there is less danger of accidentally altering a part of the file that is not being used at the time.

Other Structural Aspects of the Data Base

Using the data definition language of the new software and the dictionaries created above, this step entails designing data base views (pre-designed screens that allow the user to focus on key data elements) and access privileges (those that can be specified at the current time). Guidelines for specifying data base views include the reports produced from the existing system (see *Attachment E, Computer-Generated Reports*), knowledge of who uses which reports, and knowledge of who will need to access the various parts of the data base. For example, two data base views that will be needed immediately include the identification and receipt control fields of applications, and the status or summary of field reviews from the field review log file.

One approach in this step is to begin by determining what offices need access to various data base files, and then focusing on the key fields within the identified files. This creation of limited views serves at least three functions: (1) better data base

security; (2) simplification from the user's viewpoint; and (3) faster access to the data base.

For illustrative purposes, the following list provides preliminary specifications for access needed by various sections of OIA as well as data base files used on a regular basis:

- The DBA has complete read, write, and destroy access to the entire data base.
- The SAB has read and write access to the application and payment files, and the annual control file, and read access to other files.
- The property section of PSB has read/write access to the property file, and read-only to the other files.
- The payments section of PSB has read-only access to the batch payment file and the application files.
- The FOB has read/write access to the field report log file, perhaps the application and payment files, and read-only to the others files.
- No individuals, including the DBA, may write directly to the state level file, the batch payment file, or the application and payment history files (write access is allowed only through certain transaction programs in the system).

DATA CONVERSION

Two steps will be undertaken to convert data in the existing system to the data base in the new system. First, the data input programs in the new DDL will be created to load the new data base. This step will specify the layout of the records needed by the new data base to load in the new data, both for immediate testing and eventual conversion. These input programs will be used many times over the course of testing the new data base.

Second, the programs will be specified using new data dictionaries, input record formats, and the current file specifications to convert the data from its current files to the format required by the input programs (all described above). This will require the

DBA (or other designated person(s)) to map the old variables (sometimes from more than one file) into the new. The use of COBOL for this task would make the work faster because the old COBOL record descriptors could be used for the input sections of the programs (which are sometimes quite lengthy).

The programs developed in this phase will, essentially, do the following: read old file(s), select fields wanted, create new output records (with new fields if wanted), and write new file(s). Then, the input programs of the DBMS can read these intermediate files to load the data base. For instance, the new state level file will take data from the previous two files -- state accounts receivable and state control ledger. (See *Attachment R* in the Appendix, entitled *Mapping of Data From Old to New Files*, for a guide to this process.)

FUNCTIONAL SPECIFICATIONS

The functional specifications of the OIA computer system consist of the processes or transactions that form the basis of the system. This step involves planning and laying out the transactions that are required for OIA. The choice of an alternative system will, to a certain extent, affect the conduct of this step, primarily in the way the transactions will operate. The mainframe/microcomputer system will require more intervention by the DBA. Some transactions will be more complex because they must often be entered on the microcomputers, uploaded, then entered into the mainframe data base. Many edit checks cannot be performed on the microcomputers because the needed files do not reside there.

Having chosen a system alternative, a determination must be made regarding the actual conduct of the transactions. Choices include:

- o entirely and directly by OIA personnel (mainly for the interactive transactions),
- o with help or intervention from the DBA (for most transactions), or
- o entirely by the DBA (for batch only).

The *Description of Transactions With New OIA System (Attachment L)*, *File Implementation in a Mainframe/Microcomputer System (Attachment P)*, and the transaction descriptions in the text (p. 28-36) are available resources for making these determinations. Based upon these decisions, final specifications for each transaction can be made concerning:

- o input forms needed,
- o temporary transaction files and their structure (where needed),
- o reports resulting from the transactions, and
- o personnel procedure descriptions for each transaction.

Input Forms

For batch only transactions, the input forms needed can be re-designed (new applications, primarily). For other transactions, interactive front-ends (the user interface) will have to be designed, in light of which person(s) will perform the transaction. (See *Attachment L.*)

Temporary Files

Temporary transaction files and their structure, if needed, should be specified at this time. With the mainframe/microcomputer system, a slightly larger set of transaction files will be used to aid the interface between the microcomputers and the mainframe.

As an example, *Attachment J. Example of Interactive Data Corrections*, contains a hypothetical session of interactions between the user and the data base for updating or correcting applicant files from a TR21 form. Using this resource, this step would require the following:

- o the list of new transactions,
- o the map of each transaction to various files (see *Attachment R. Mapping of Data from Old Files to New*, and *Attachment I. Transactions and Computer Files*),
- o the new file layouts (created above), and

o the results of the transactions.

Further, each condition leading to an error must be specified, as well as information needed from the user to accomplish the transaction. If the transaction is done in batch, any errors specified must be listed in the report which is generated with that transaction.

Reports

A list of reports must be specified, focusing on reports that are either part of periodic maintenance (weekly, monthly, annual) or those to be made available on an ad-hoc basis. This list will, in effect, be a modification of the reports currently available (see *Attachment E, Computer-Generated Reports.* An on-line inventory of available reports should be created as well as programs that allow users to select reports from menus. This activity should be undertaken with an eye to the new data base views because in many cases the ability to quickly view parts of the data base may eliminate the need for reports or reduce the frequency of the requests.

Personnel Descriptions

A set of personnel procedure descriptions corresponding to each transaction should be established. The descriptions will include error checking routines to be followed in the OIA office and the specific duties of each person involved. For example, every time a property analyst makes a change, a second individual should review the changes; all field report update transactions should be channeled to one person in the field office (even though many field officers may update the field report log file regularly). The reliance on the DBA for particular functions related to the transactions should also be completely specified by this point in the process.

MAINTENANCE

There are several specified maintenance procedures to be developed and implemented as part of regular system upkeep, both on a weekly and annual basis. Regardless of which alternative system is chosen, the weekly schedule will include:

- o backing-up the entire system to tapes,
- o performing any transactions that are the responsibility of the DBA,
- o compiling statistics on data base usage,
- o running data base integrity-checking program(s), and
- o generating weekly reports.

In addition, if the mainframe/microcomputer system is being implemented, a few additional tasks will need to be performed, probably on a weekly or bi-weekly basis. These tasks include: (1) uploading of receipt control and field report data, and (2) downloading of new payment computations and their associated data files or fields. These two functions should be tied as closely as possible to batch processing, whether it is done weekly or more frequently.

For the weekly tasks of compiling statistics on data base usage, a small file and program should be developed to keep track of the transactions and queries done on the data base so that the frequency of use of various fields and files can be analyzed. This process will provide input for modifying the data base organization, adding keys and indices, and designing new reports in the future so as to optimize the system's functioning and increase its usefulness. For example, useful statistics include:

- o for each transaction -- frequency, time of day, time to complete;
- o frequency of access of each file in the data base;
- o size of various files in the data base as a function of time of year;
- o number of users -- simultaneous, per day, week; and
- o frequency of requests for various reports.

The data base integrity programs may be run weekly or less often, depending on the necessity. They will insure the data consistency and quality control of the data base. By checking these on a regular basis, most logical anomalies created in the data base can be discovered quickly and corrected.

The annual maintenance schedule to be established will reflect the yearly cycle of applications, allocations, and payments which OTA performs. This cycle can be specified in terms of the transactions against the data base and must adhere to the following order:

1. system purge and archive,
2. annual start-up,
3. annual file initialization,
4. checkpoint transactions as incoming applications are received,
5. addition of new applications and preliminary application transaction,
6. initial allocation transaction, and
7. initial payments transaction.

After these transactions have been performed, others may occur (for a specified fiscal year) in any order desired, as the need arises.

DEVELOPMENT AND TESTING

In this step, an overall plan for system development and testing is conceived. These processes should be viewed as iterative in nature -- develop a program (or series of programs) to perform a specified transaction, test the program, refine the program, develop the next program(s) to perform another transaction, test, etc., until all the pieces are in place.

An incremental approach to development and implementation is proposed, as follows:

- o choose a file,

- o specify the transactions against that file,
- o design the interactive front-end,
- o design the reports to be generated from the transaction,
- o begin to write, and
- o test.

During the period in which one file is being tested, another can be in the design process. A schedule of development for the files, the transaction programs, and the corresponding reports should be compiled. This method offers two advantages. First, program modules developed for one transaction can be used in others, thereby integrating the system of programs as much as possible. Second, it is easier to locate program bugs if the series of programs are built and tested incrementally. Files that are building blocks for other, more complex processes should begin first, and files that are relatively independent of others can be started at any time. This bottom-up or incremental approach to system development is a safe and reliable method that centers around defining a new file, writing transaction program(s), getting some test data into the new file, and testing the transaction(s).

One technique for testing is loading some current data from the existing system into the new test file and running transactions against it, at the same time that they are run in the existing data base. The results of the two processes can be compared to help find errors or problems. Other examples of test steps in this development/test process include:

- o dumping small subsets of data, using the programs written above for conversion for use in the tests,
- o building small data base files which have the same structure as the final ones but only a few fields,
- o when an entire file is designed, testing it with all fields but only a small subset of data, using the data loading programs above,
- o running typical requests on each file to test the ad-hoc queries and programs as they are written and de-bugged,

- o when testing transaction programs which access more than one data base file, beginning with only a few fields in each file,
- o structuring the tests, going over every transaction and within each one, every possible flaw or illegal data value or data type possible, and
- o using the listings of error conditions in the current "SAFA User's Guide" for ideas and keeping track of new errors discovered during this process to be added to the error code master listing in the system documentation.

A specific example of this development/testing process is provided for the new property file. The tasks to be performed are specified as follows:

- o design the new property file,
- o define that part of the new data base in the DDL,
- o write the data base program to load the data into the new property file,
- o write the program to convert the data from the old Property 1 file to the new property file,
- o write the user front-end to perform property file updates,
- o define one or more views of the new property file as desired,
- o dump a small amount of data from the active Property 1 file in the data base for testing,
- o begin to compile a list of error codes and their meaning,
- o load some data into new data base,
- o begin testing the update program (transaction #2),[1]
- o enter all possible types of erroneous data and make sure the program handles them, and
- o try all possible functions -- add a new record, delete a record, and update fields on an existing record.

Development and testing of the overall set of files and their corresponding transactions should proceed in the following order:

---

[1]The reference to transaction numbers can be found in *Attachment L* in the Appendix.

(a) property file - transaction #2

(b) field report log file - transaction #10

(c) ID file - transaction #1

(d) state file, batch payment file - transaction #5

(e) application and payment files - transactions #3, #12, #13, and #15

(f) application and payment history - replica of application and payment file, and then transaction #11

(g) design of continuing application shell file (replica of main application file), and new applicant request file, then test transactions #3, #4, and #7

(h) design of disaster and construction files, and transactions for them

(i) design of annual system purge (transaction #14) and weekly backups (transaction #16)

The final phase of system testing will consist of simulating the complete data base, first with separate files and finally with the entire data base. When all the obvious problems are removed and all the files and transactions perform properly, the current data base can be unloaded to separate files, from which the data base is then loaded into the new system, using the programs written previously. At this point the new system should be put through a final battery of tests for a period of perhaps 1 to 2 months. During this period, every person in OIA who will use the new system should perform typical interactions using the test system. Since the OIA system consists of many small, frequent interactions, it will likely be impossible to actually maintain two concurrent data bases, both completely accurate. However, it is not necessary at this step to have all the data in the test data base be accurate and up-to-date, only that the full volume and range of data be tested. Once again, tests should be structured by transaction and by every possible data combination within each transaction.

DOCUMENTATION

Two types of documentation will be necessary -- one for the DBA that is technical in nature and one for the user community (OIA personnel). The first type of documentation will contain such items as the following:

- o data dictionaries for all files in the data base,
- o procedures followed to perform the annual maintenance,
- o procedures followed for weekly or other regular maintenance of the data base,
- o the location of programs (source code) and their documentation, files used to compile and link-edit them, and location of executable files,
- o error codes from various transactions and their meanings,
- o use and updating of the annual constants file,
- o description and use of the file for assessing the performance of the data base,
- o methods for creating, altering, or destroying data base views,
- o files and passwords needed for altering other users' access,
- o log of weekly and annual maintenance performed,
- o upkeep of the reports available, and
- o any other technical details needed by persons managing the system.

The second type of documentation will be a user's guide, available to anyone in OIA needing to access and use the system. It will include such as information as:

- o how to sign-on to the system,
- o how to access the various data base views and browse through the data,
- o inventory of available reports and how to obtain them, and
- o pre-canned transactions available and detailed instructions for performing them.

PHASING-IN

The development processes described in the previous sections, shown on the timetable in *Attachment Q*, will take place over approximately a 3-year period in an iterative fashion. Most of the processes will overlap and reinforce each other, and the order implied by the chart reflects primarily the order of start-up. By the time most of the testing of the new system is complete, final plans for phase-in should be made. The phase-in period should be planned to coincide with a slow time of the year so that there is more opportunity to iron out problems that may still occur. Final training of staff should occur prior to the phase-in of the new system.

The phasing-in of the system should take a short period of time, perhaps two weeks, and involves the following steps:

- the new software system is emptied of all test data,
- current transactions are halted,
- the entire data base is dumped, one file at a time, and
- the new data base is loaded.

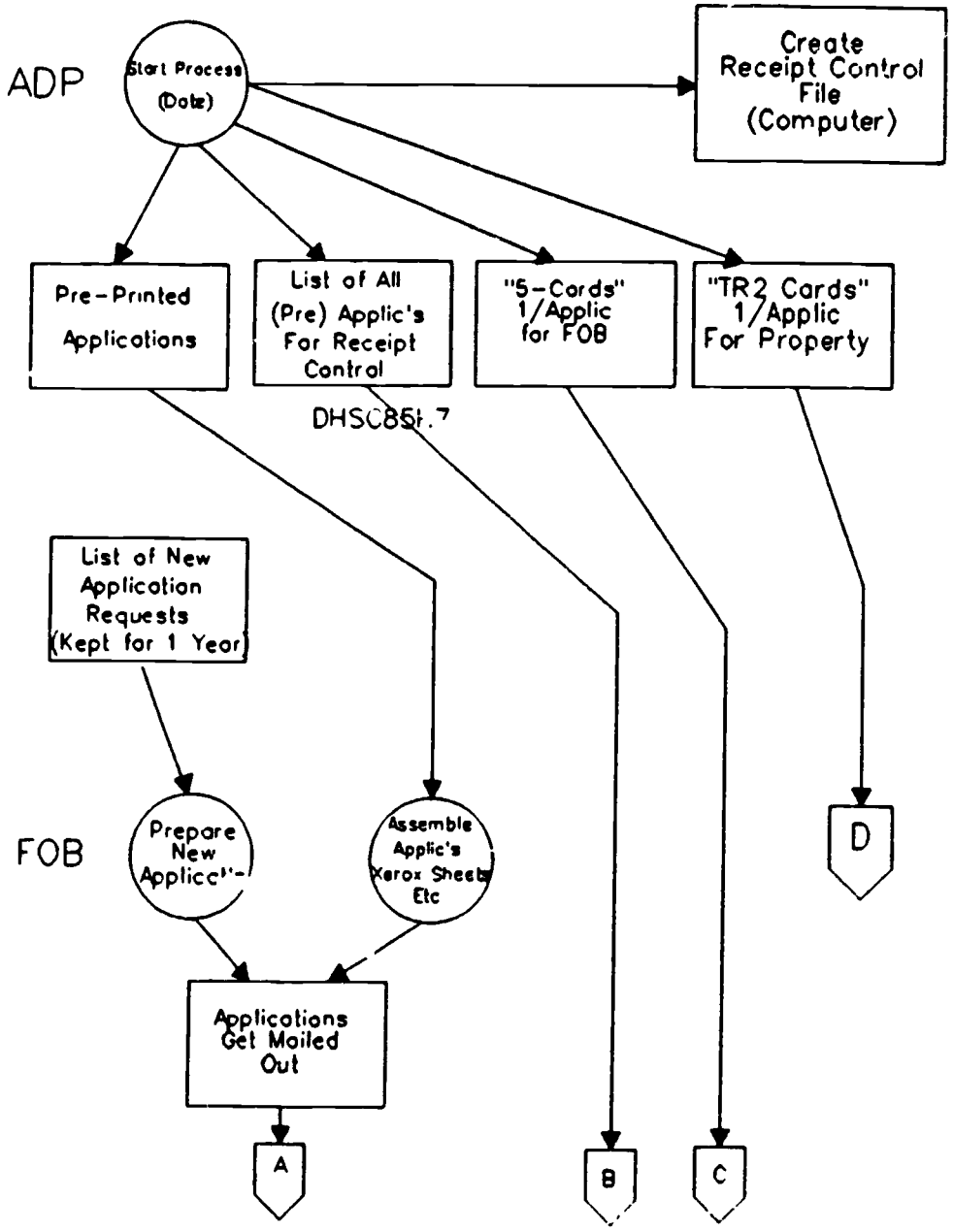
Files that are completely new (disaster, construction, field report log) do not need to be loaded at this time (with the exception of the annual constants file).

Additional features can be added to the system at any time. For example, if a dial-up modem and dedicated PC were installed as part of the OIA system, it would be a simple matter for field officers to call in and send their field reports, once the forms for this process were designed. Similarly, application input forms could be sent on floppy diskettes to the LEAs where they would be filled out and mailed back. They could then be loaded in and merged with the continuing applicants shell file to become a new year's application.

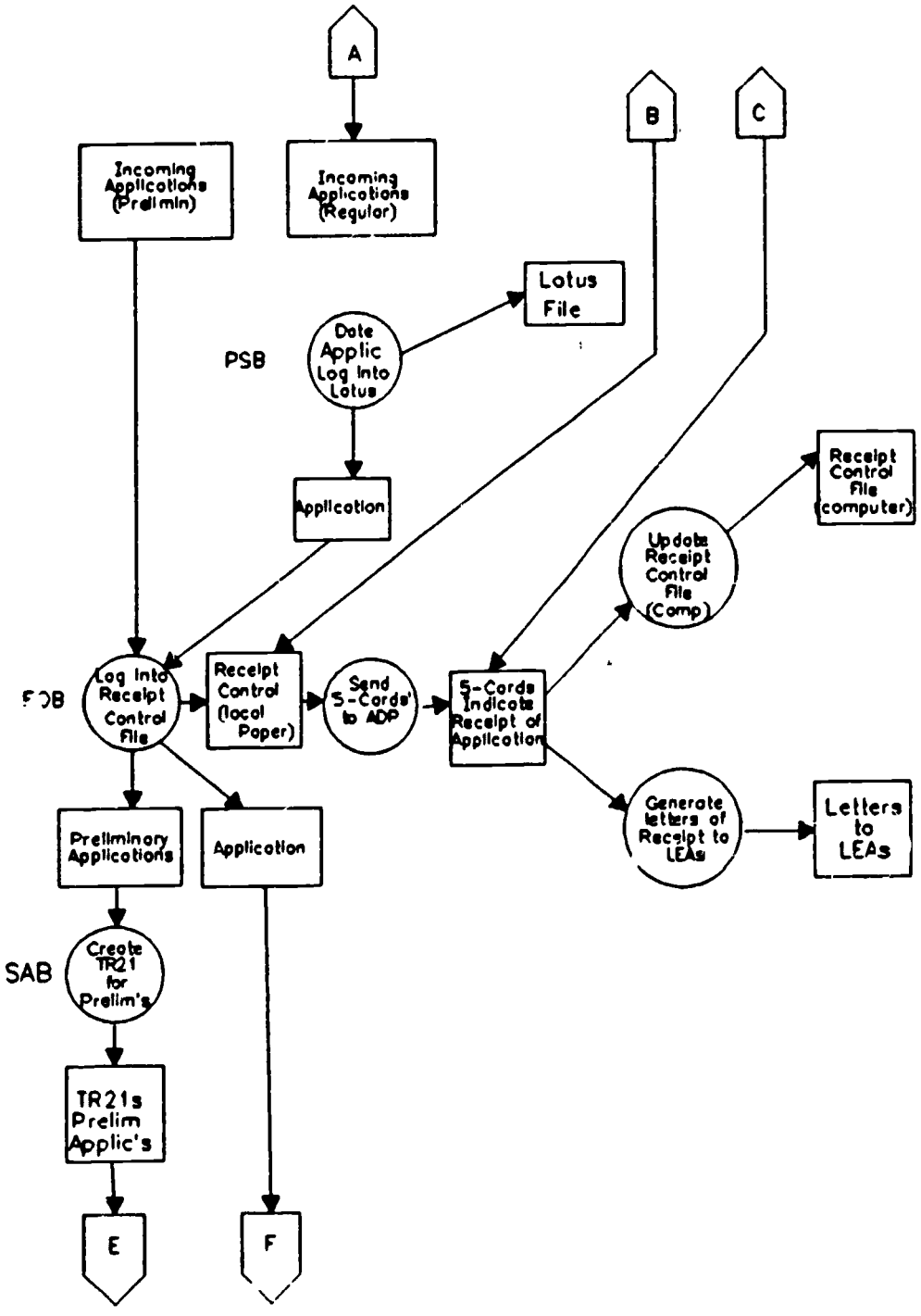
APPENDIX

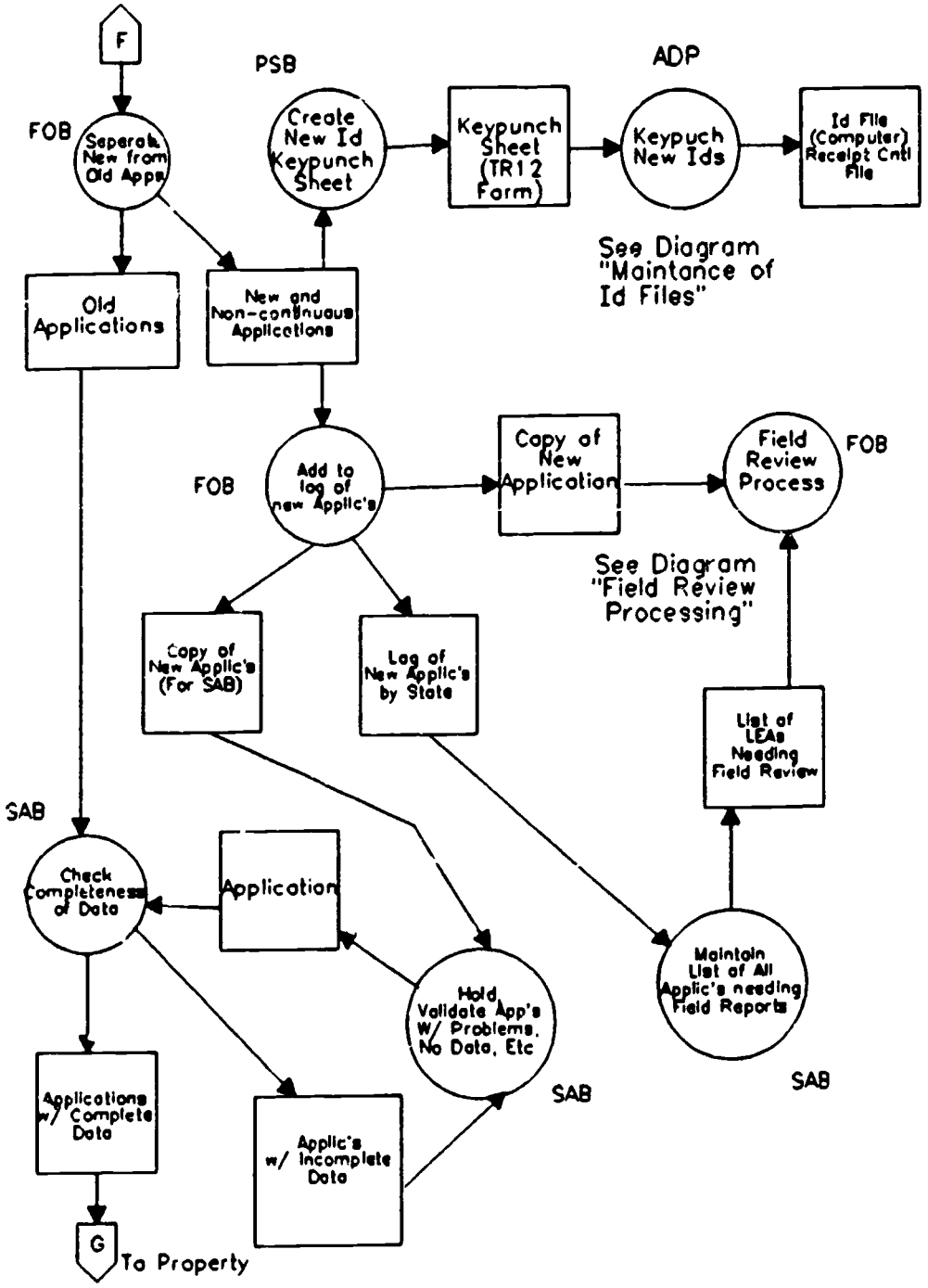
ATTACHMENT A

APPLICATION TO PAYMENT PROCESSING

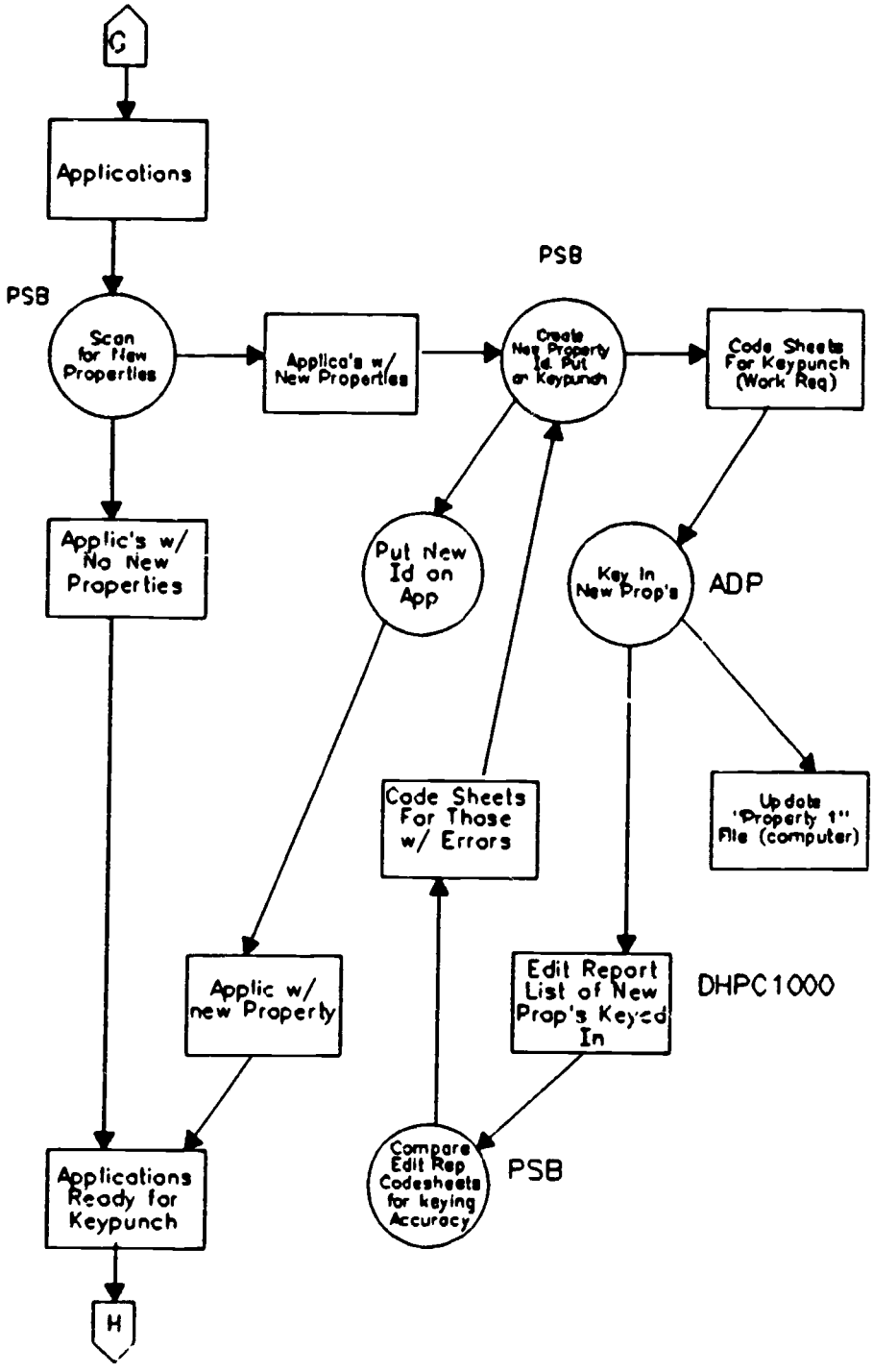


Attachment A (continued)

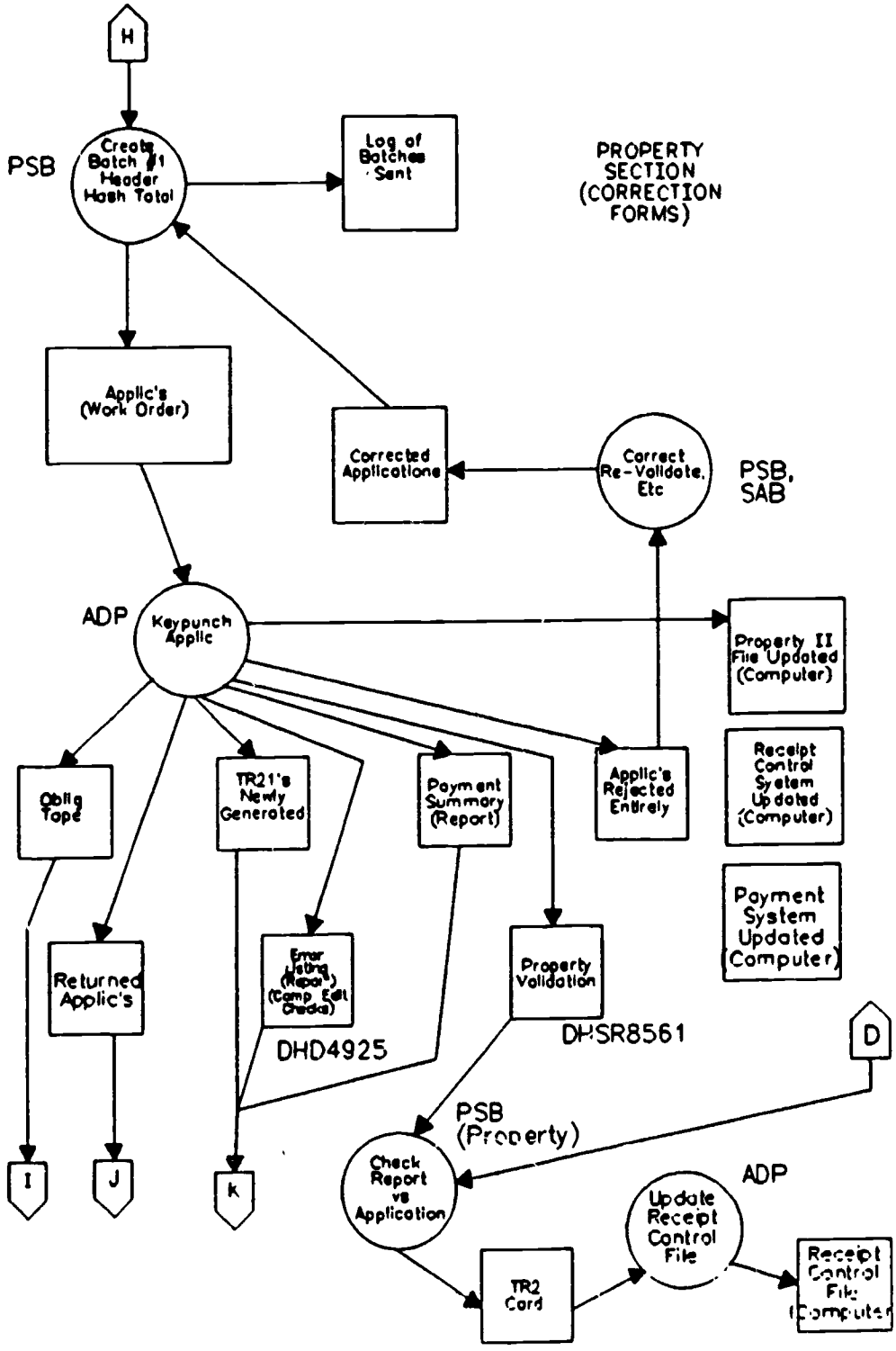



F
PSB
ADP
FOB
Seperate New from Old Apps
Create New Id Keypunch Sheet
Keypunch Sheet (TR12 Form)
Keypuch New Ids
Id File (Computer) Receipt Cntl File
See Diagram "Maintance of Id Files"
Old Applications
New and Non-continuous Applications
FOB
Add to log of new Applic's
Copy of New Application
Field Review Process
FOB
See Diagram "Field Review Processing"
Copy of New Applic's (For SAB)
Log of New Applic's by State
List of LEAs Needing Field Review
SAB
Check Completeness of Data
Application
Hold Validate App's W/ Problems, No Data, Etc
Maintain List of All Applic's needing Field Reports
SAB
SAB
Applications w/ Complete Data
Applic's w/ Incomplete Data
G
To Property

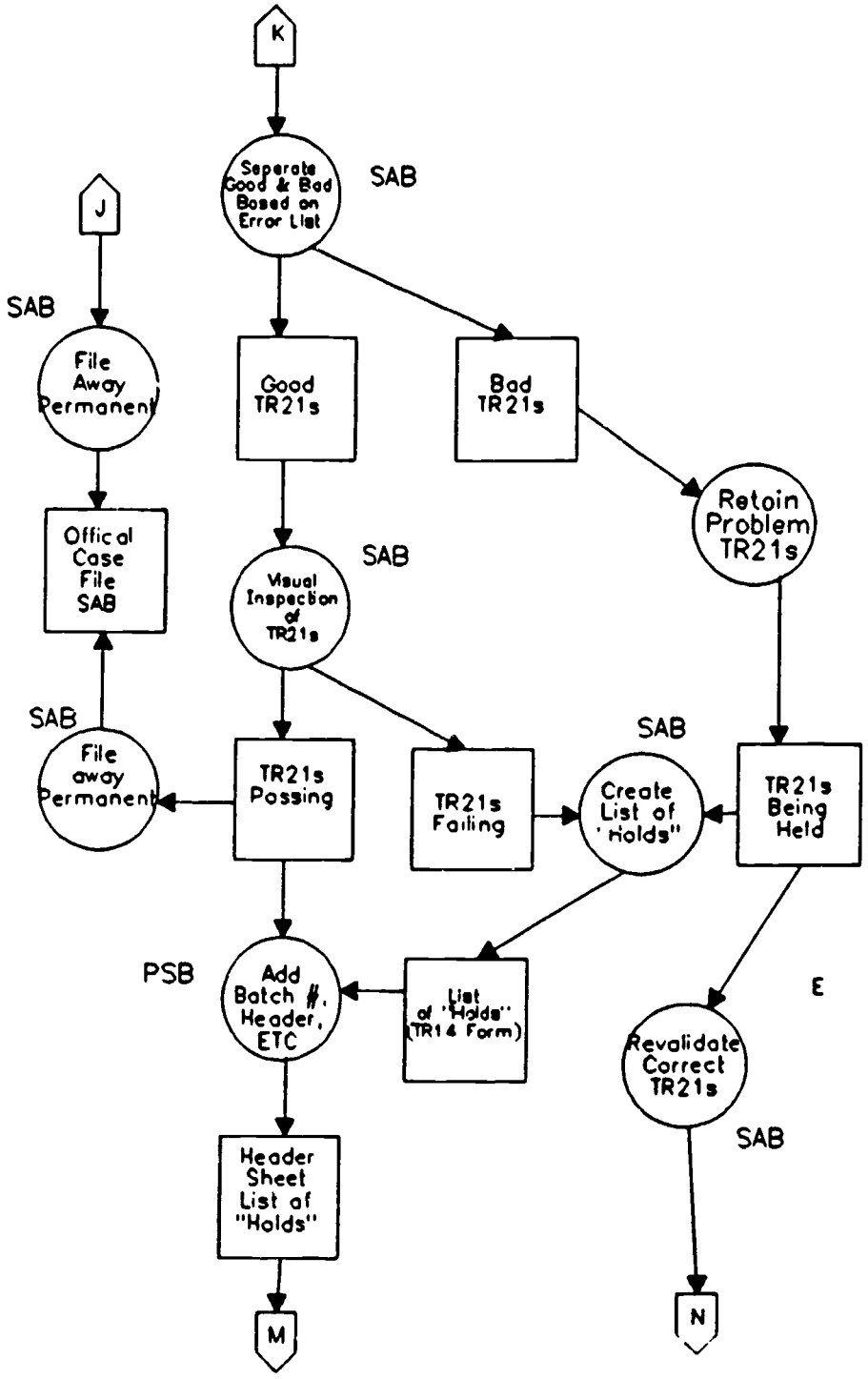
Attachmcnt A (continued)

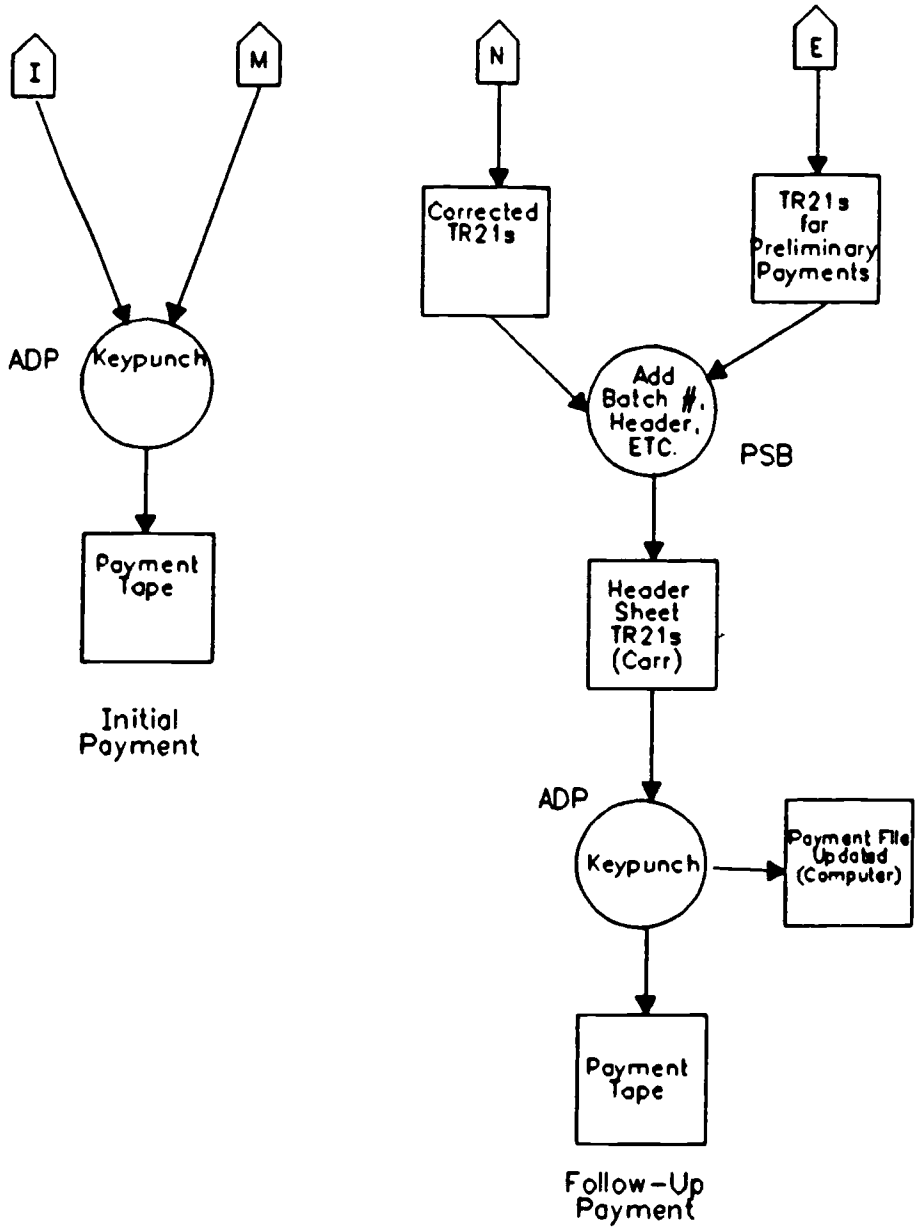


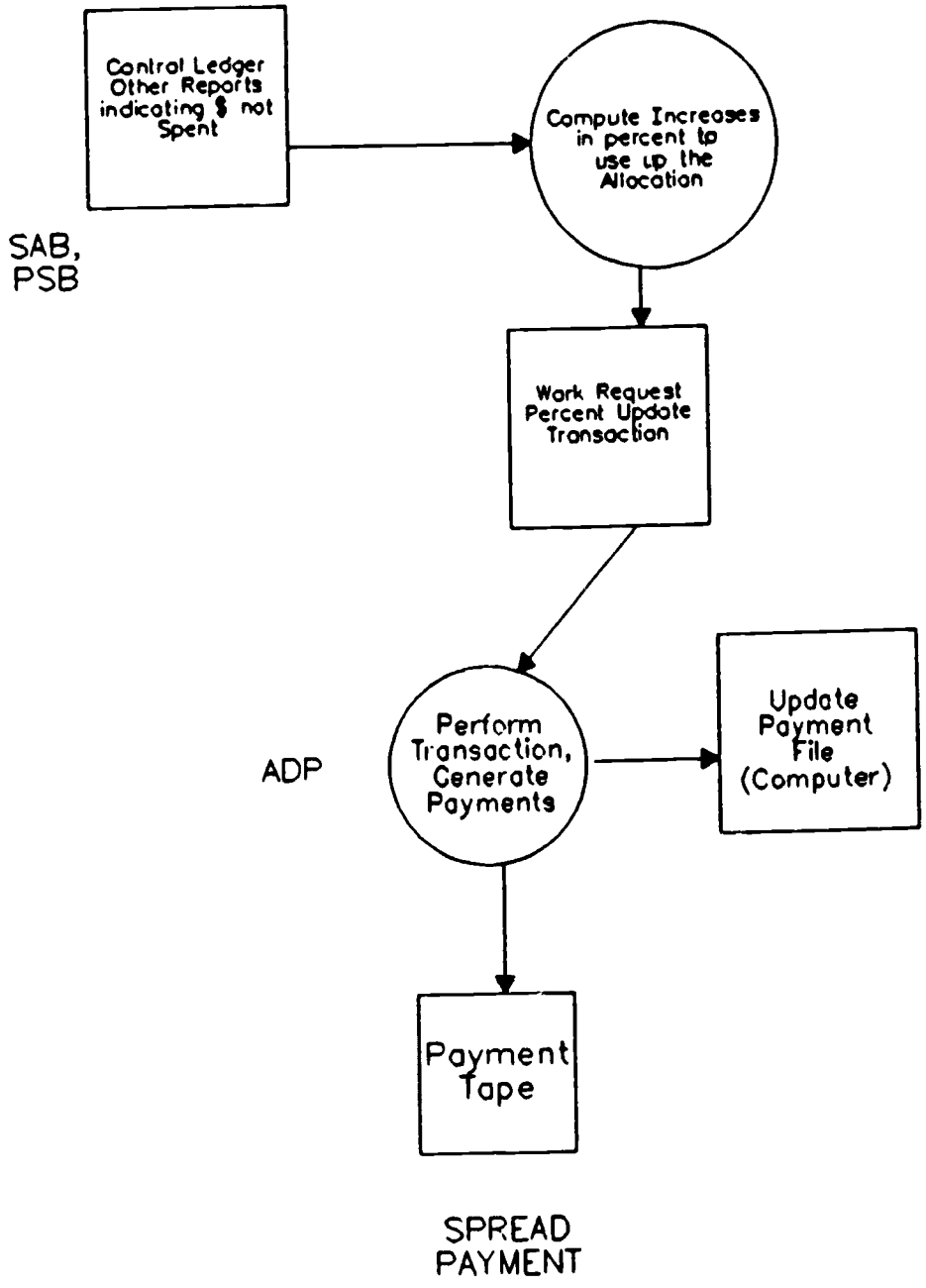
Attachment A (continued)



A-5

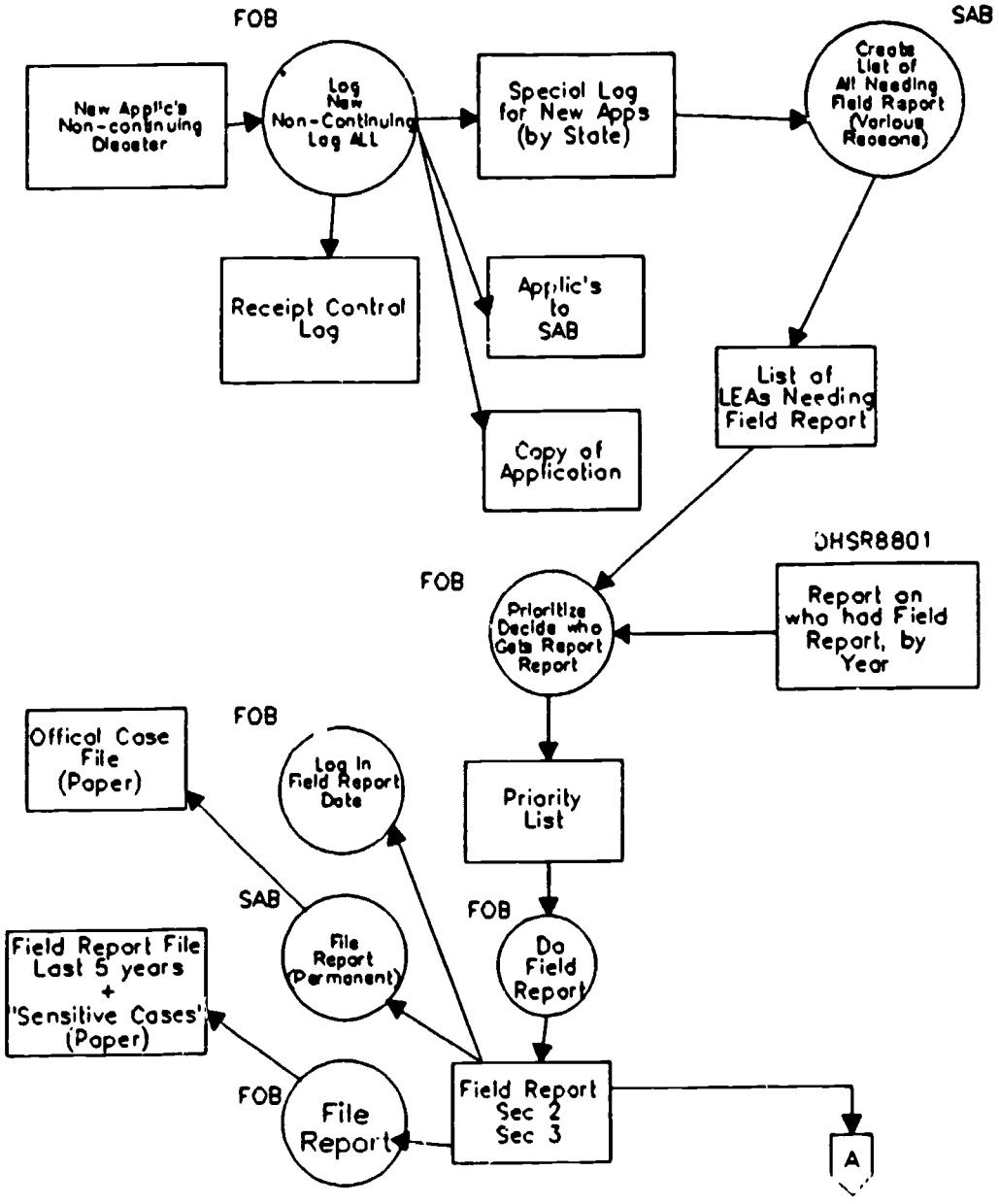

K
Separate Good & Bad Based on Error List
SAB
J
SAB
File Away Permanent
Good TR21s
Bad TR21s
Retain Problem TR21s
Official Case File SAB
Visual Inspection of TR21s
SAB
SAB
File away Permanent
TR21s Passing
TR21s Failing
SAB
Create List of "Holds"
TR21s Being Held
PSB
Add Batch #, Header, ETC
List of "Holds" (TR14 Form)
E
Revalidate Correct TR21s
SAB
Header Sheet List of "Holds"
M
N


I
M
ADP
Keypunch
Payment Tape
Initial Payment
N
E
Corrected TR21s
TR21s for Preliminary Payments
Add Batch #, Header, ETC.
PSB
Header Sheet TR21s (Carr)
ADP
Keypunch
Payment File Updated (Computer)
Payment Tape
Follow-Up Payment


Control Ledger Other Reports indicating $ not Spent
Compute Increases in percent to use up the Allocation
SAB, PSB
Work Request Percent Update Transaction
Perform Transaction, Generate Payments
ADP
Update Payment File (Computer)
Payment Tape
SPREAD PAYMENT

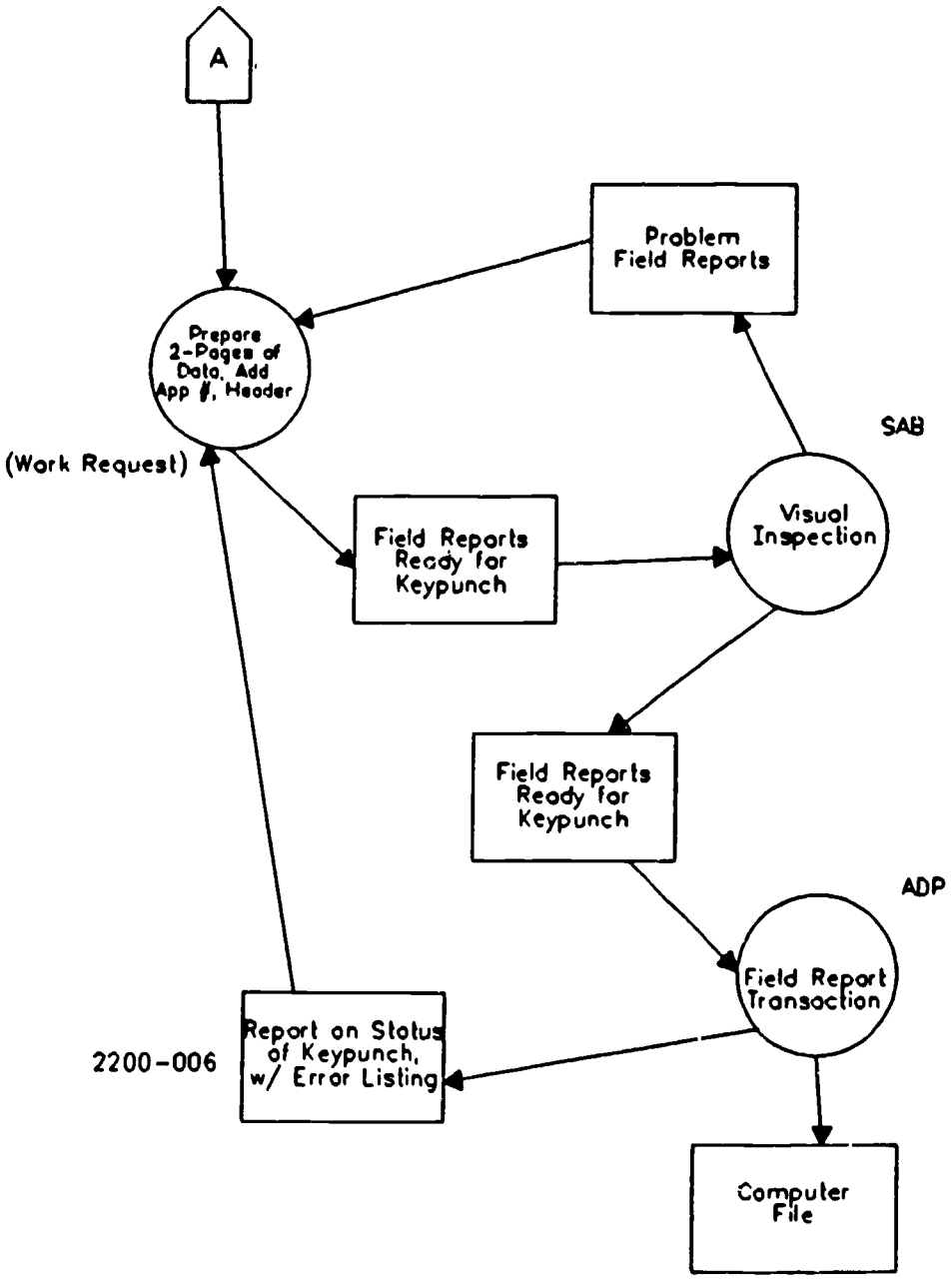
ATTACHMENT B

FIELD REPORT PROCESSING



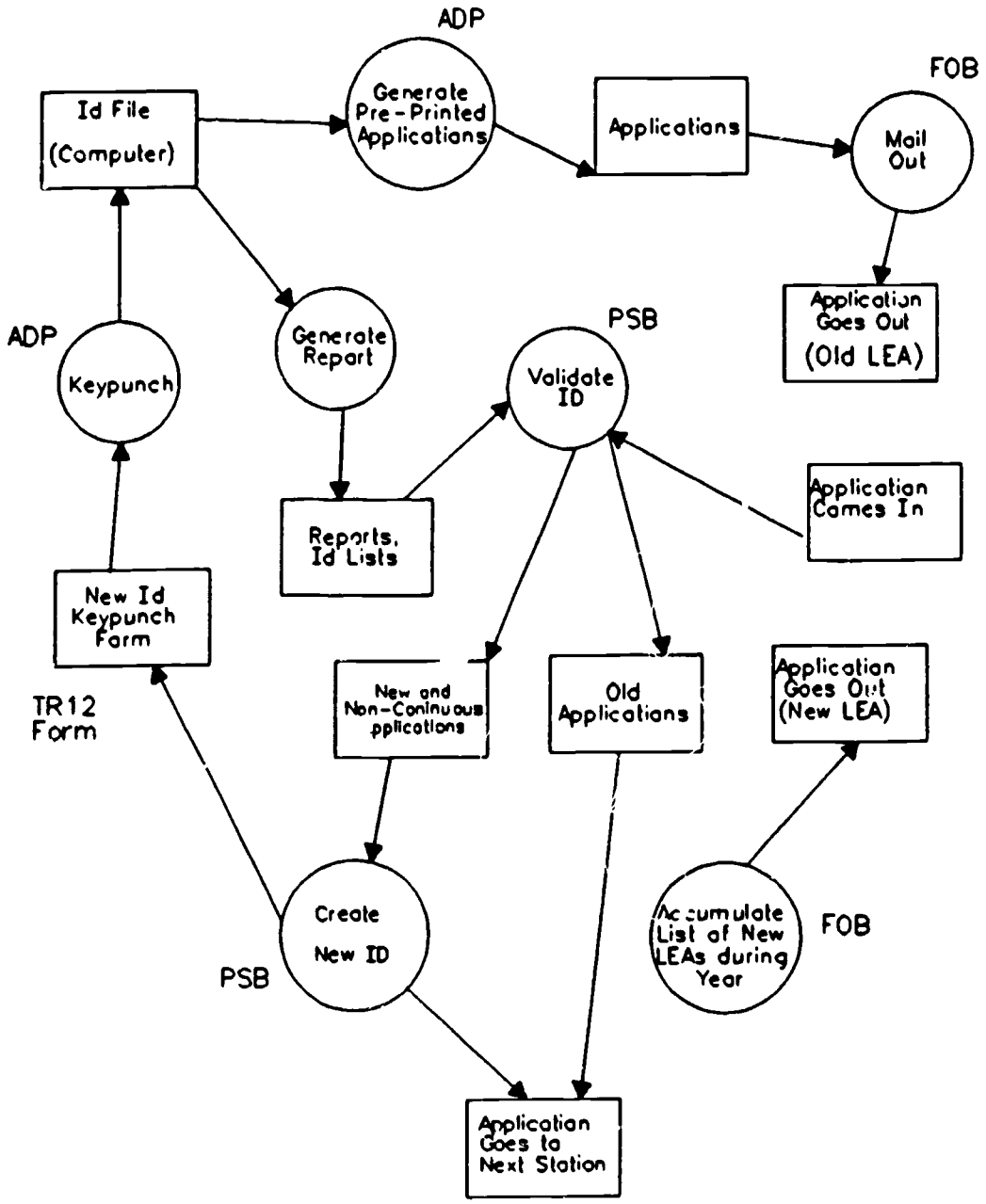
B-1

Attachment B (continued)



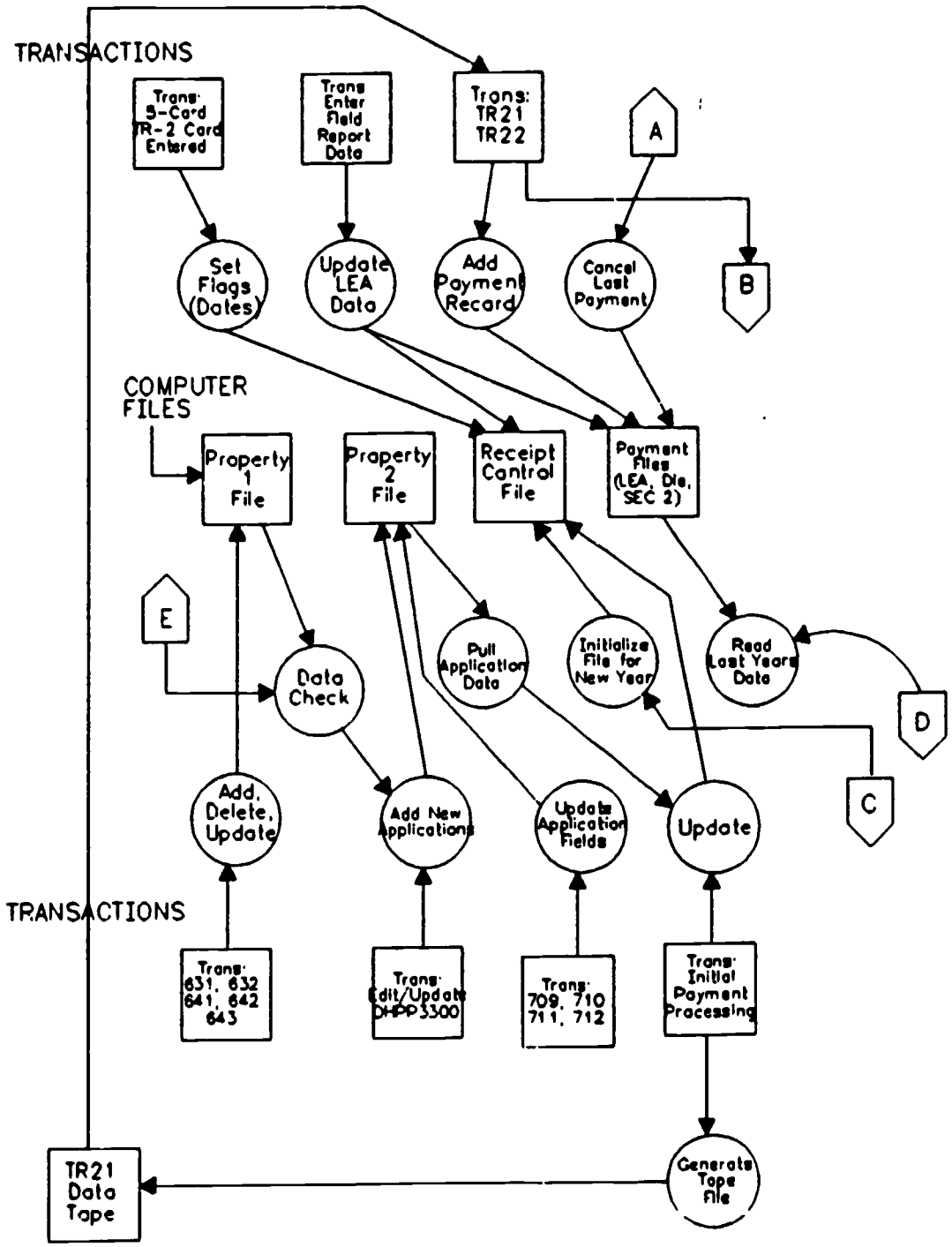
ATTACHMENT C

MANAGEMENT OF ID (LEA) FILE



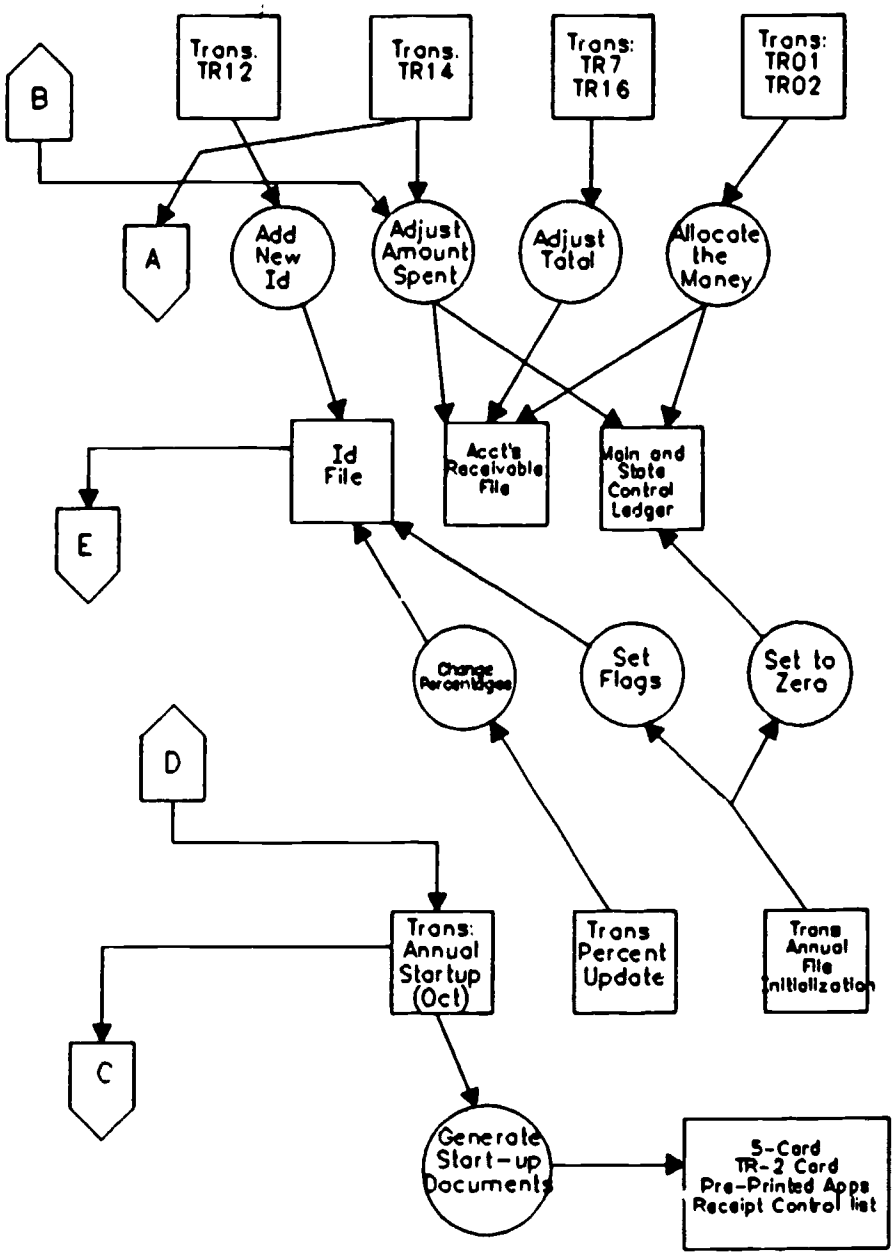
ATTACHMENT D

TRANSACTIONS AND COMPUTER FILES



D-1

Attachmcnt D (continued)



ATTACHMENT E

COMPUTER-GENERATED REPORTS

Annual Start-up Processing:[1]

Field Office Application and Reports Log -- Pre-printed applications to send out -- "5-cards" used by FOB indicating receipt -- "TR2 cards" used by property indicating OK	(DHSC85H7)

Initial Payment Processing: (DHDP4925)

TR21 forms	
Summary of Entitlement (form 4119, 4119A)	(DHD1600, DHD1650, DHD1700)
Output tape (obligation tape)	
Notification of Grant in Aid Action (form 424)	(DHD1550)
Congressional Notifications Report (OE4194)	(DHD1500)
TR21 Selection Listing	(DHD4925-001)
Error Listing	(DHD4925-002)
Payments Listing	(DHD4925-003)
Reject Listing with reasons	(DHD4925-004)

Batch Processing: (DHD01)

New applications

SAFA Edit/Update	(DHDP0100)
Property Validation List	(DHSR8561-001)

TR21 transactions

Batch Payment summary	(DHD2350-001)
Transaction Error List (data dump)	(DHD1425-001)
Processing Summary Batch list	(DHD1450-001)
Transaction Error list	(DHD1450-002)
ID file list, supplemental	(DHD1350-001)
Transactions Changed to Estimates and Reason	(DHD1150)
Payment voucher	(DHD1700)
Less than $5,000 Rejection Letters	(DHD1800-001)
Schedule of Payments	(DHD1300-A)
Million Schedule	(DHD1300-B)
Deleted Payments	(DHD1300-C)
Applicant accounts receivable list	(DHD1400)

[1]Where available or applicable, the code number of the report is given in the right column.

Other transactions

Batch back-ups	(DHDP0150)
Batch Reports	(DHDP0125)
Processing Percent Update Summary	(DHD0050-001)
DSA Input Transactions Entered	(DHD1050-001)
Record Processing System transaction error list	(DHD1100-001)
Accounts Receivable Update Activity	(DHD1200-001)
Financial Management Info. Systems Trans list	(DHD1250-001)

Weekly Processing: (DHDP0200)

State Accounts Receivable Report	(DHD2150-001)
Entitlements and Payments Report	(DHD2200-001)
List of worksheets to be generated	(DHP2200-002)
Receipt Control	(DHD2300-001)
Receipt Control	(DHD2300-002)
Section 2 Status Report	(DHSC8592)
Section 3 Status Report	(DHSC8549)
Main Control Ledger*	(DHD2050)
State Control Ledger*	(DHD2100)
M & O Ditto Report*	(DHD3350)

Monthly Processing: (DHDP0300)

Accounts Receivable History	(DHD3050)
Summary Balances of Accounts Receivable	(DHD3150)
Monthly Transmittal Accounts Receivable	(DHD3200)
Section 7 Obligations, Payments, Activity	(DHD3400)
Applications Received and Processed	(DHD3500-001)
Cash Payments	(DHD3600-001)
Unliquidated Obligations	(DHD3650-001)
Funds Obligated	(DHD3650-002)
Obligations and Expenditures	(DHD3650-003)
Use of Allocated Funds	(DHD3650-004)
3d2B Report	(DHD3625-001)

Annual Processing: (DHDP0400)

Table 1	(DHDC4100-001)
Table 3	(DHDC4150-001)
Table 5	(DHDC4200-001)
Total ADA and TCE	(DHDC4250-001)
Unliquidated Obligations	(DHDC4300-001)
Accounts Receivable	(DHDC4400-001)
Actual Obligations	(DHDC4450-001)
Expenditures and Accounts Receivable	(DHDC4500-001)
Table 2	(DHDC4550-001)
Expenditures and Accounts Receivable	(DHDC4650-001)
FMIS Annual Report	(DHDC4700-001)

*Also produced monthly.

Table 15	(DHDC4800-001)
Labels	(DHDC4750-001)
County Book	(DHDC4825-001)
Table E	(DHDC4850-001)
5 Year History of Applications Received	(DHDC4875-001)
SAFA Information Retrieval Systems	(DHPC2450)

Other Processing: (periodic, on request)

Record of Field Reports, 5 years	(DHSR8801)
Property Validation list, short form (with FOB no.)	(DHD4920-001)
Applicants not in Final Pay Status	(DHS85A5-001)
Schedule 9 Quarterly Report (financial statement)	
ID file list, complete	(DHD1375-001)
Summary Allotment Main Control Ledger	(DHD2050-001)
Maintenance & Operations Branch Control Record	(DHD2250-001)
Applicants (> 1975)	(DHDL1475)
Districts with SPED ADA	(DHSC8858-001)
Report of Eligible Properties Claimed, by state	(DHPC1300)
Report of Properties Claimed, by state	(DHPC1275)
Property File/transaction activity/error list	(DHPC1000)
Section 8 Low Rent Housing Pupils	(DHSC8831)

ATTACHMENT F

LOGICAL DATA GROUPS WITH KEYS

The following list identifies, from a logical point of view, the computer files necessary for implementation of the proposed system(s). The name of each file is given, along with a short description of what it contains and the keys used to access the file. Note that there are five separate files associated with application and payment data. In addition to those listed here, files equi·alent to the currently used report extract files could be added, if this makes regular report generation easier. Further, temporary input files can be used as needed (described elsewhere) for implementing OIA input verification systems.

ID Files	Uniquely identifies every LEA ever in the system. Used to identify an LEA across years. Contains general information about the district, also has status information on the past five years: what types of aid applied for and received, how much, field report done, and a few others. Approximately 40 fields, 8,000 records. keys: **ID (11 digits)** **EIN #**
Property Files	Uniquely identifies every federally-owned property that has ever been claimed on an application. Does not contain any history, only the most updated information on the property. Approximately 30 fields, 40,000 records. keys: **PROP ID (13 digits)**
Application and Payment Files (Main Component)	The main component is unique within each year for each LEA. If an LEA qualified under two sections of the law, it would have only one main component, but could have two or more separate sectional record types corresponding to those sections of the law applied for (see below). Main component has identifiers; receipt control data (20 fields) such as date(s) approved and type of aid applied for; accounts receivable data, if any (overpayment information for the current year) (5 fields); payment summary data when actual payments are calculated; and data up to 20 transactions (assuming there are never more in a given year), such as date and payment computed. keys: **ID FY**
Application and Payment Files (Section 2, Main)	Application and payment components depend upon which sections of the law applied under. For every Section 2 applicant, there will be one record per applicant per year, containing: Table 8 Data, Table 9 Data, TR22 computed data, and Section 2 payment data. (See *Attachment H. Variables Needed for Application and Payment Data*). keys: **ID FY**

Application and Payment Files (Section 2. PROP)	In addition, every Section 2 applicant will have a file containing only Table 10 data, with one record per property claimed. keys: **ID FY PROPERTY**
Application and Payment Files (Section 3, Main)	Section 3 applicants will have, first, a record containing: Table 6 Data, Table 7 Data, TR21 computed data, and Section 3 payment data. keys: **ID FY**
Application and Payment Files (Section 3, PROP)	In addition, Section 3 applicants will have records in a separate file containing one record per property claimed, with Tables 1 through Table 5 data. Note that this logical storage scheme is computer efficient, but the actual paper application will remain the same; that is, Tables 1 through 5 will continue to be entered on separate sheets. keys: **ID FY PROPERTY**
State-Level File	State-level records, one per year, per program type, per state (number of applications, amount of entitlements, amount spent), plus accounts receivable (state), and the state control ledger data (total of no more than 40 fields). keys: **ID STATE FY**
Batch Payment File	Batch payment summary: unique records for each batch, containing number of payments or other actions (summary counts, date). This only has 10-15 fields, and as many records as batches per year. keys: **FY BATCH #**
Annual Constants	This file contains annual constants, i.e., total appropriation, percent entitlements, "edit windows." Could also hold the main control ledger information. keys: **FY**
Congressional File	This file will be the same as the one currently used, containing such information as name, address, and district number. keys: **STATE CONG. DISTRICT**
CRS File	This file remains the same as the current one, used as a link to other files in the MIDAS system, as needed. keys: **EIN #**

Field Report Log File	This file is new, containing information about every field report done in a given year. More detail on field reports could be entered here, even if only a few fields are eventually used for correcting main application file(s), as is now done. keys: **ID FY**
Continuing Applicants Shell File	This is a new file, which would in many ways replicate the applicant and payment files (above), and would reflect data printed on the pre-printed application. Its primary use would be to serve as a template for entering continuing applicant data. When annual start-up is performed, prior year data are used to create this file. This is then used to create the next year's applicant file, thereby avoiding re-entering data. Once deadlines for applications are past, this file could be eliminated. keys: **ID FY**
New Applicant Request Log File	This would be a new file, containing basic information about all LEAs which request applications for the first time. This file would be used to track requests. It could also be used to help generate new applications for those LEAs requesting them, at the start of each year, or as a template for entering the full application later. keys: **ID (if assigned) FY**
Applicant and Payment History	This file would be identical to the main applicant and payment file above. It would contain records for every change that occurred in the main files. It would thus contain a complete record of application data changes; along with the main file, a complete six year record would always exist. keys: **ID FY**
Transaction Files	In addition to the above, three new temporary files, used for interactively performing transactions, would be added to the data base. These consist of: a) TR21 transaction file, used for editing and final visual scan when initial payments are made, before main files are updated; b) follow-up payment file, also used for editing payment calculations before finally applying them to the payment files; and c) preliminary application file, for those LEAs requesting preliminary payment before their complete application is received. In all three cases, as soon as the transactions are approved and run against the payment files, these temporary files are eliminated; they serve as a substitute for or enhancement of paper forms currently used.

All other data found on reports and applications can be obtained by either aggregating up to the state, year, or national levels, or by joining several of the above files using the identifiers as match variables and/or selectors.

ATTACHMENT G

CALCULATION OF FILES SIZES FOR IMPACT AID SYSTEM

The calculation of sizes and structures of files needed is based directly on two things: the current computerized system and the preceding listing of logical data groups. The logical data groups, in turn, derive directly from the system requirements, both current and new, as described. For each file needed, a short description is provided, followed by a calculation of the size of storage needed, the structure of the file, and approximate size and number of records to be expected. For the purposes of designing a data base and comparing differing implementations of the proposed data base, rough estimates are included of how often each file will: 1) have a record added, 2) have a record deleted, 3) have a record updated; or 4) have a record accessed and read. Predicting these frequencies is necessarily approximate, in particular for the lookup portion, because this depends upon reports requested throughout the year.

APPLICATION AND PAYMENT DATA

This file would contain all data for the application from each LEA, plus the receipt control data, the accounts receivable data, and a few flags for each of up to 10 transactions for the year. Since some basic data is kept for all LEAs, this portion is constant; the section of the law applied under determines which other types of records, and how many of each, exist for a given LEA.

<u>All applications</u> (MAIN COMPONENT, 3,000 applications)

3,000 * 10 numeric variables
\+ 1 string
\+ 20 fields for receipt control data
\+ 5 fields for accounts receivable data
\+ 20 variables, payment summary data =

3,000 * 260 bytes = **0.78 MB** total size
3,000 observations, 260 bytes each

add new record:	3,000/year
update record:	30,000/year
delete record:	never
look up record:	60,000/year

Section 2 only (300 applications)

MAIN COMPONENT
300 * 100 numeric variables * 4 bytes/variable = **0.12 MB** total size
300 observations, 400 bytes each

add new record:	300/year
delete record:	never
update record:	100/year
look up record:	1,000/year

PROPERTY COMPONENT
300 * 30 variables * 4 bytes/variable * 50 properties = **1.80 MB** total size
15,000 observations, 120 bytes each

add new record:	15,000/year
delete record:	1,000/year
update record:	3,000/year
look up record:	5,000/year

Section 3 only (2,700 applications)

MAIN COMPONENT
2,700 * 103 variables * 4 bytes/variable = **1.12 MB**
2,700 observations, 412 bytes each

add new record:	2,700/year
delete record:	never
update record:	900/year
look up record:	9,000/year

PROPERTY COMPONENT
each observation consists of:

13 bytes (property) + 20 bytes (5 count fields) = 33 bytes
2,700 * 33 bytes/observation * 50 props = **4.45 MB**
135,000 observations, 33 bytes each

add new record:	135,000/year
delete record:	never
update record:	30,000/year
look up record:	50,000/year

3D2B only (3D2B COMPONENT)

300 * 63 variables * 4 bytes/variable = **0.08 MB**
300 observations, 250 bytes each

Total size needed = 0.78 + 0.12 + 1.80 + 1.12 + 4.45 + 0.08 = **8.35 MB** approximately for every year on line. This calculates to approximately 8 * 6 years, or near **50 MB** total for the application and payment files.

PROPERTY FILE

This file would remain essentially the same as the current Property 1 file, the main change being the reduction of a few unneeded fields.

40,000 observations * 40 fields * 4 bytes/field = 6.4 MB total size
40,000 observations, 160 bytes each

add new record:	500 new records/year
delete record:	100 deletions/year
update record:	300 updates, corrections/year
look up record:	20,000 properties claimed per year
	20 references for validity check or to print per property =
	500,000/year total, concentrated in 2-3 months

ID FILE

This file remains conceptually similar to the current ID file, except that it is simpler, and contains only one type of record, consisting of data directly associated with the LEA: Annual constants and variables are now moved to the new "annual file" (see below), and control fields indicating status of the LEA are moved to the application and payment files (above).

8,000 total LEAs ever * 50 fields * 4 bytes/field = 1.6 MB approx. size

add new record: 200 new records/year
delete record: rarely
update record: 400 updates or changes/year
look up record: 3,000 (applications)/year

+ 5 reports/year = 15,000/year

ANNUAL STATE LEVEL FILE

This would be one file with observations for every state, program type, and year. The fields would include state and program summary data, such as number of applications, amount of entitlements, amount spent, etc. (20 fields) plus state accounts receivable data (5 fields), plus the state control ledger data (15 fields).

(60 States) * 4 program types * 6 years of data * 40 fields * 4 bytes / field = 0.25 MB approximately.

add new record:	180/year
delete record:	never
update record:	200 batches/year * 60 states * 3 program types = 36,000/year
look up record:	for reports only; 9,000 / year

BATCH PAYMENT DATA

This file would contain data pertinent to the batch processing, particularly payment processing. It would have one observation generated for every batch throughout the year, with other data such as number of payments or other actions, summary counts, date, batch number, appropriation, CAN number, obligation, total paid, etc. This is the same data as the current main control ledger.

200 batches/year * 6 years * 20 fields * 4 bytes/field = **0.10 MB**

add new record:	200/year
delete record:	never
update record:	never
look up record:	for reports only; 10,000 times/year

ANNUAL CONTROL DATA

This file would contain all those fields, constants, etc. which are unique for each year. There is no current file which is equivalent, because many of these fields are now embedded in the computer code. Fields such as annual appropriation for all three sections, percent entitlements, currently active edit windows, and other annual constants would increase OIA's ability to keep track of and control the payment processing. These could be arranged so that programs which compute payments could read data from this file, thereby rendering current parameters more accessible. Logically, this file has one record per fiscal year and could be stored as a spreadsheet for easy viewing and/or editing, as long as the format was fixed so that the programs doing the computations for payments could reliably read it.

This file would be very small, perhaps 5 to 10 kilobytes only, so size calculations are not important. The important aspect is that it is instantly accessible and editable by the SAB.

CONGRESSIONAL DISTRICT FILE

This file is the same as that currently existing; it contains Congressional district code and name, state, and name and address of the current representative, necessary to generate the letters at the beginning of each year. It would only contain the current status; no history is necessary.

550 districts * 100 bytes each = **0.06 MB approximately**
550 observations, 100 bytes each

Add new record:	150 every 2 years
delete record:	150 every 2 years
update record:	occasionally, 50/year
look up record:	550/year

HISTORY OF APPLICATIONS AND PAYMENTS

This file has an identical structure as the main applications and payments file described above, that is, it actually consists of 1 main file and 4 smaller ones, depending upon the application. This file is used to store the history of data used for payment calculations, if that data pertaining to any one fiscal year ever changes after payments have been calculated at least once. Therefore, the main file serves as a history for those LEAs with no changes; when changes do occur, all data pertaining to the payment calculation is written to this history file *before* the main file is updated for further calculations. Changes in the data are possible either because TR21 transactions have been made, or field office reports have been used to correct the data. Since the main purpose of this file is keep a legal record of the basis of all payment calculations, this file is not needed often. Therefore, it does not necessarily have to be stored in as accessible a form as the main files. See the text section on "file integrity" for further explanation of this file.

Assuming that roughly 1,000 applicants per year have these types of changes made, and that all six years data are kept in the same file, the total size of these files would be approximately one-third that of the main application + payment file, or approximately 20 MB.

add new record:	1,000/year
delete record:	never
update record:	never
look up record:	200/year

FIELD REPORT LOG FILE

This is a new file used to increase the level of automation of the field report processing (see p. 8-9). It consists of any data correction fields now used (child counts, LCRs, etc), as well as a series of receipt control fields to help keep track of field reports and their progress. It would have records added throughout the year as field reports are performed and data are generated from them. The data correction fields would be used to supply data when field report update transactions are performed. For a given year, the file would end up with approximately 2,000 records, one per report.

2,000 reports * 30 data correction fields (now used)
30 receipt control type fields (new fields)

2,000 * 60 numeric fields * 4 bytes/field * 6 years = **2.88 MB total size**

add new record:	2,000/year
delete record:	never
update record:	10,000/year
look up record:	20,000/year

CONTINUING APPLICANTS SHELL FILE

This file is new and would essentially reflect data sent out on the pre-printed application which is sent every year to all continuing applicants. It would not be necessary to retain this file beyond the application deadline every year. As explained in the text, this would be the template allowing the re-keying of application data quickly and easily and with less chance for error.

For size calculations, the size of the main applicant file is used minus the variables on payment summary, TR21 and TR22 data (that is, any to-be-computed fields). Assuming that there are 2,500 (total) continuing applications sent out every year, and taking only one year's data, the calculation yields:

MAIN COMPONENT:

2,500 * 10 numeric variables
+ 1 strings = 2,500 * 80 bytes = **0.20 MB**

Section 2 only (300 applications)

MAIN COMPONENT
200 * 63 numeric variables * 4 bytes/variable **0.05 MB** total size
200 observations, 252 bytes each

PROPERTY COMPONENT
200 * 30 variables * 4 bytes/variable * 50 properties = **1.20 MB** total size
10,000 observations, 120 bytes each

Section 3 only (2,300 applics)

MAIN COMPONENT
2,300 * 25 variables * 4 bytes/variable = **0.23 MB**
2,300 observations, 100 bytes each

PROPERTY COMPONENT
each observation consists of:

13 bytes (property) + 20 bytes (5 count fields) = 33 bytes
2,300 * 33 bytes/observation * 50 props = **3.80 MB**
115,000 observations, 33 bytes each

3D2B only (3D2B COMPONENT)

200 * 63 variables * 4 bytes/variable = **0.05 MB**
200 observations, 250 bytes each

Total size needed, one year only =
0.20 + 0.05 + 1.20 + 0.23 + 3.80 + 0.05 = **5.53 MB**

NEW APPLICANT REQUEST LOG FILE

This file contains information about new or non-continuing LEA's which request information on the Impact Aid program throughout the year. It would be used to generate new applications at the start of the fiscal year's processing, because a few fields of data could already be determined and entered.

This file could contain perhaps 40 fields plus name, address, etc., for a total of approximately 300 bytes per record. The file would grow throughout the year as requests came in, up to perhaps 500 observations.

500 records * 300 bytes/record = **0.15 MB**

add new record:	500/year
delete record:	150/year
update record:	500/year
look up record:	2,000/year

TRANSACTION FILES

These files are temporary, and are used to allow careful visual inspection of payments calculated before these are made permanent on the payment files. Therefore, their existence is short, but space must be provided for them as follows:

TR21 Transaction file

250 records/batch * 103 field * 4 bytes/field =	**0.10 MB**

Follow-up payment file

approximately the same as above =	**0.10 MB**

Preliminary applicant file

250 records/batch * 25 fields * 4 bytes/field =	**0.03 MB**
Total space needed = 0.10 + 0.10 + 0.03 =	**0.23 MB**

In summary, the total size of the data base can be estimated as follows:

	size	components
Application + Payment	50.00	6
Property	6.40	1
ID file	1.60	1
State Level file	0.25	1
Batch Payment file	0.10	1
Annual Control file	-	1
Congressional file	0.06	1
Application and Payment History	20.00	6
Field Report Log file	2.88	1
Continuing Applicant Shell file	5.53	6
New Applicant Request file	0.15	1
Transaction files (temporary)	0.23	3
	87.20 MB	29 components

New fields and even new files may be needed in the future, therefore, none of these figures can be exact until data dictionaries are specified. Since there is some space overhead needed for data base definitions (dictionaries, record descriptors, indexes, etc.), the above figure is multiplied by 25 percent, yielding a space requirement of approximately 109 megabytes for the entire data base.

ATTACHMENT H

VARIABLES NEEDED FOR APPLICATION AND PAYMENT DATA

ALL APPLICATIONS (3,000)

IDS, ETC.
10 numeric vars
10 strings (names, etc.)

RECEIPT CONTROL DATA
20 variables/app

ACC. RECEIVABLE DATA
5 vaiables/app

PAYMENT SUMMARY DATA
10 vars/app (totals, payments)

SECTION 2 (300)

TABLE 8 DATA
30 vars/app

TABLE 9 DATA
33 vars/app

TABLE 10 DATA
30 vars/app/prop
(average of 50 props/app)

TR22 COMPUTED DATA
20 vars/app

SECTION 2 PAYMENT DATA
10 vars/app

SECTION 3 (2,800)

TABLES 1 - 5, each
prop ID + count/prop/app

TABLE 6
16 vars/app

TABLE 7
7 vars/app

TABLE 8 (3D2B only, 300)
30 vars/app

TABLE 9 (3D2B only, 300)
33 vars/app

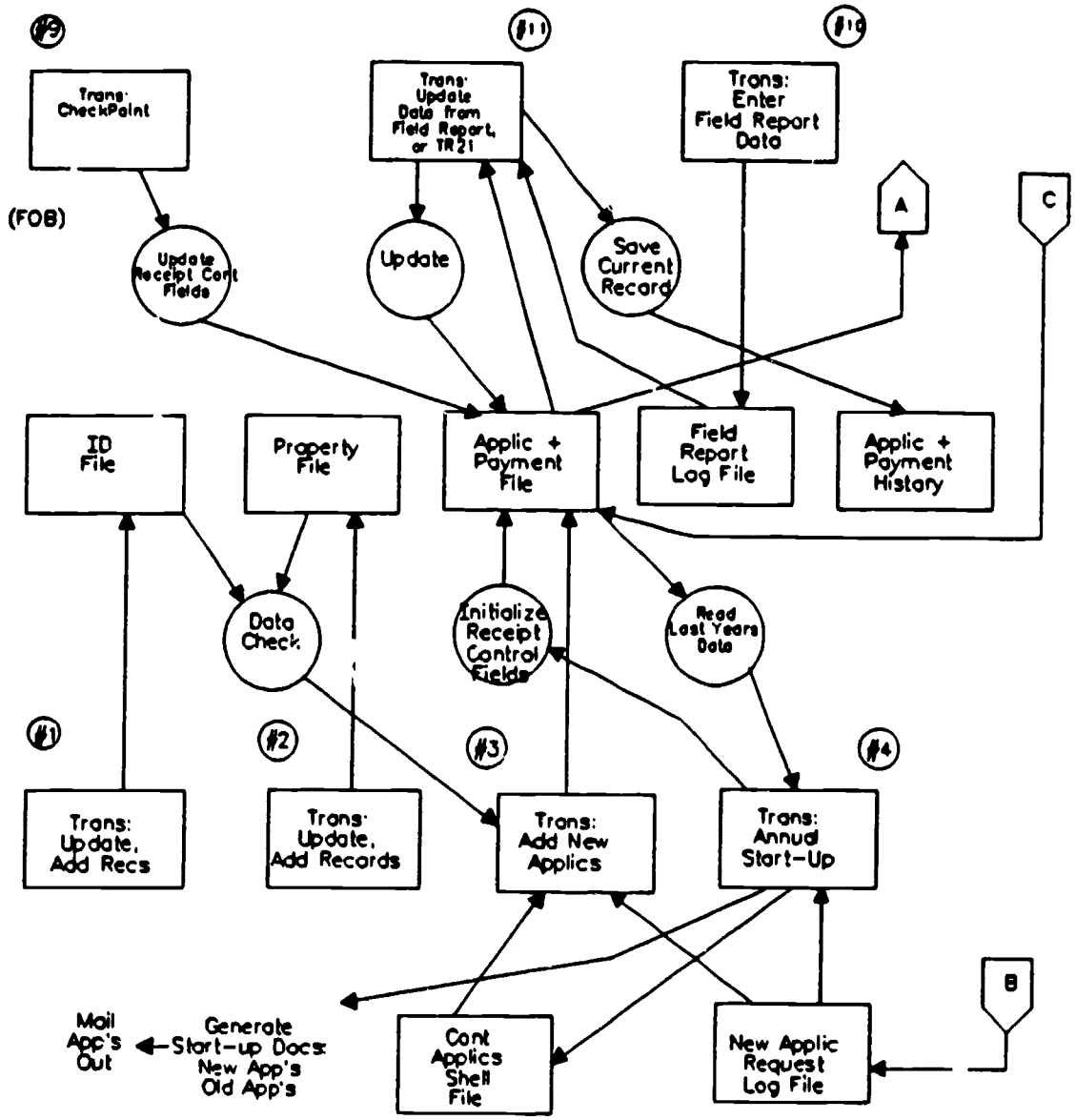
TR21 COMPUTED DATA
20 vars/app

SECTION 3 PAYMENT DATA
20 vars/app (% of LCR)
20 vars/app (prorate %s)
20 vars/app (sums, totals)

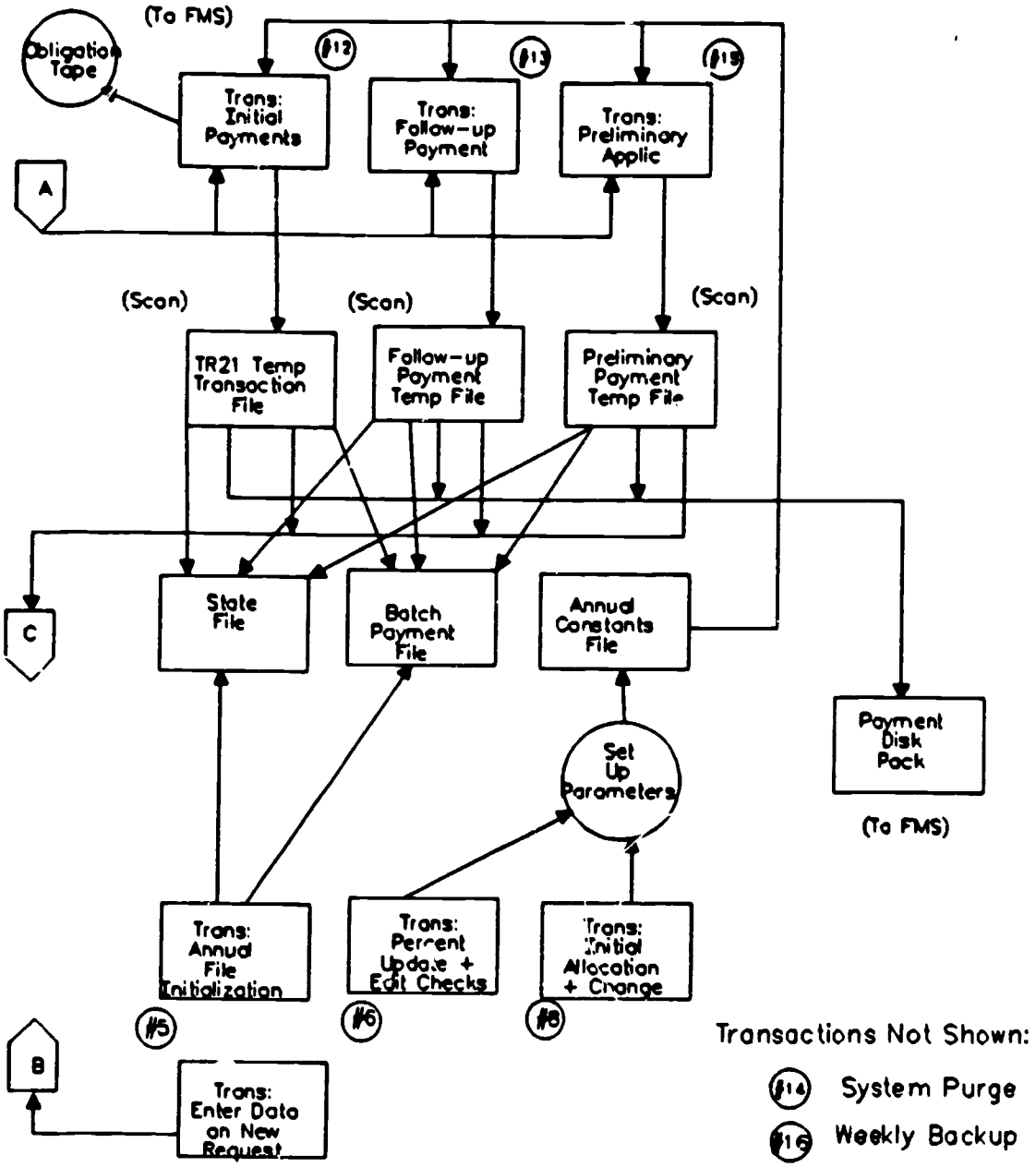
Note: Tables 1 - 5 and Table 10 have as many records as properties claimed; a given property may appear in any combination, or in all, of these tables. A given property may also be claimed by more than one applicant.

ATTACHMENT I

TRANSACTIONS AND COMPUTER FILES, NEW SCHEMA



Attachment I (continued)



ATTACHMENT J

EXAMPLE OF INTERACTIVE DATA CORRECTIONS

The following is a hypothetical example of a sequence of prompts and responses illustrating the kinds of interactions that might occur when correcting the application and payment files. This will be possible through the development of user-interfaces, as described in the text. Such a session might take two minutes (less for an experienced user) and is completely controlled by the user.

USER: RUN CORRECTION PROGRAM
MACHINE: ARE YOU CORRECTING FROM A FIELD REPORT OR VIA A TR21?
USER: TR21
MACHINE: IS THE APPLICANT SEC 2 ONLY, SEC 3 ONLY, OR BOTH?
USER: SEC 4
MACHINE: SORRY, NO SECTION 4. MUST BE SEC 2, SEC 3, OR BOTH.
USER: SEC 3.
MACHINE: WHAT IS THIS YEAR'S ID FOR THE APPLICANT?
USER: 15-FL-88-2039
MACHINE: APPLICANT YOU HAVE SELECTED IS:
CENTRAL SCHOOL DISTRICT
1 MAIN STREET
DADE COUNTY
MIAMI, FLORIDA 21432
APPLICANT ID FOR 1988:15-FL-88-2039

THE FOLLOWING SCREEN WILL CONTAIN THE CURRENT DATA FOR THIS APPLICANT. PLEASE USE THE CURSOR AND NUMBER KEYS TO CHANGE ANY FIELDS NECESSARY. WHEN FINISHED, MOVE THE CURSOR TO THE BOTTOM LINE AND TYPE "END."

(SCREEN FILLS, USER TYPES CHANGES, THEN TYPES "END.")

MACHINE: YOU HAVE ENTERED CHANGES FOR TABLE 5 DATA, MEANING THAT PROPERTY DATA MUST BE CHANGED. WHAT FOLLOWS ARE THE PROPERTY ID(S) FOR TABLE 5 CURRENTLY CLAIMED. PLEASE CORRECT THE APPROPRIATE ONE(S). WHEN FINISHED, MOVE THE CURSOR TO THE BOTTOM LINE AND TYPE "END."

(SCREEN FILLS, USER SCROLLS DOWN LIST OF PROPERTY IDS WITH CHILD COUNTS. USER CHANGES TWO OR THREE, THEN TYPES "END.")

MACHINE: YOUR CHANGES HAVE BEEN SUMMED, AND MATCH THE TR21 DATA WHICH YOU HAVE INDICATED. DO YOU HAVE ANY MORE CHANGES TO MAKE TO THIS APPLICANT?

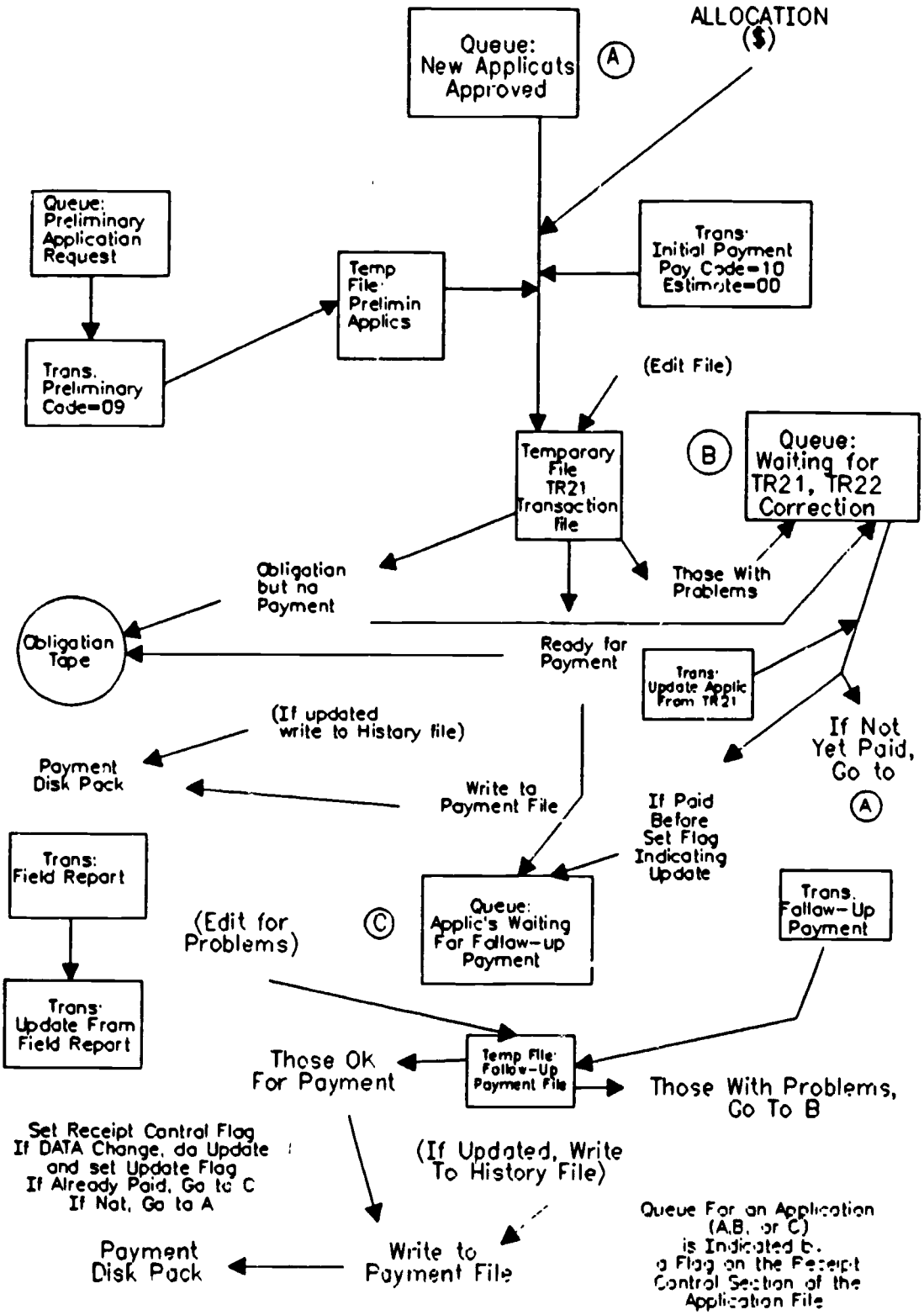
USER. NO

MACHINE: DO YOU WANT A PRINTED LISTING OR AN ON-SCREEN LISTING OF CHANGES ENTERED?

USER:	ON (LISTING OF OLD AND NEW VALUES SCROLLS ON THE SCREEN).
MACHINE:	WRITE THE CHANGES TO THE TRANSACTION FILE? (Y/N)
USER:	Y
MACHINE:	FILE "TR21UPDT.FEB1989" HAS BEEN CREATED. DO YOU WISH TO MAKE MORE CORRECTIONS NOW? (Y/N)
USER:	N
MACHINE:	FILE "TR21UPDT.FEB1989" IS READY FOR UPDATING, AGAINST MAIN PAYMENT FILES.

ATTACHMENT K

SCHEMATIC OF BATCH PROCESSES



ATTACHMENT L

DESCRIPTION OF TRANSACTIONS WITH NEW OIA SYSTEM

The terms "batch" and "interactive" are frequently used in data processing, but often take on somewhat different meanings. Actually, few computer and data processing tasks are entirely batch or entirely interactive, because the concept really represents a continuum. In general, the term "batch" refers to the degree that transactions or processes are pre-packaged or "canned," and thus pre-determined; consequently, it is the degree to which the user cannot interfere with or alter a process once it has begun. As a result, "batch" usually refers to tasks taking place in the background; that is, tasks which are not directly under the control of or even visible to the programmer. The extent to which a process is canned usually indicates how much of a batch process it becomes. Also, "canned" programs limit what the end-user is permitted to do on the data base.

The transactions conceived in the proposed alternative system are, to varying degrees, "canned" such that there is control over (1) specific data that can be changed, (2) processing that can be performed, and (3) who is allowed to perform the transactions. However, there is some variation in the degree of direct user control across the different transactions. For example, transactions 1, 2, 3, 7, 9, and 10 will consist of interactively updating fields in some of the data base files by the various OIA offices; transactions 6 and 8 consist of interactively editing the annual constants file and changing or adding parameters. On the other hand, transactions 4, 5, 14, and 16 are mostly canned because they do virtually the same thing every time, so they are batch-type programs. Finally, the complex transactions which involve computing payments, hand checking them (if desired), and updating payment files (5, 11, 12, 13, and 15) will consist of a combination of canned programs and user interfaces offering choices to be made, such as how many LEAs to process, which ones, and whether to complete the transaction or allow user checking first.

The user interfaces (such as the example in *Attachment J*) provide the means for non-technical users to control the processing as desired. Thus, OIA staff will be interactively using programs that are primarily "canned" (what the programs can do is very specifically laid out).

Since a DBMS with a fourth generation language is proposed, another type of processing with the system will be possible -- the direct, ad-hoc querying of the data base for instant information, retrieval, and report-making. It should be noted that *updating* data through this mechanism should be strictly circumscribed, if allowed at all; updating will normally be performed only via standardized transactions. This interactive querying is distinct from the standard transactions listed below, which are all, by and large, "canned."

1) Update ID file - interactive, adds new records of new LEAs and changes fields in existing ones as needed. Performed one at a time via terminal. Perhaps 200 new LEAs per year and 200 updates.

2) Update property file - interactive only, adds new records for new properties and updates data on existing ones as needed. Performed one at a time via terminal, 300 new, 300 updates annually. Another function that this transaction must do is to search the current year's application files (those that may be affected by the given property change), and print out a small report on those claiming the property changed. Then analysts can examine those applications and generate a TR21 update if called for.

3) Add new applications - a) new applications, use a batch transaction. Reads in keypunched data, and references the new applicant request log file for simple descriptive data and as a cross-check; b) continuing applicants, use new interactive transactions which read the continuing applicants shell file for a template, and add or change only fields needing it.

4) Annual start-up - this is a batch job and can be set up to be almost completely automated because each year the same thing is done. This transaction reads last year's application and payment file, creates main component of the new year's application and payment files (receipt control primarily), creates the new year's continuing applicants shell file, and generates pre-printed applications for continuing applicants. It also reads the new applicant request log file, generates application and payment file main component for these, and prints out applications for new applicants (these might only have a few fields pre-printed). This is performed once per year.

5) Annual file initialization - This transaction is a straightforward batch job, performed once a year. This transaction prepares the batch payment file for new entries and zeroes out (initializes) the state level file for the new year.

6) Percent update transaction - interactive, performed perhaps once a month. This simply consists of editing the annual constants file and changing the few parameters there. Adding, deleting, or changing the edit checks in force can also be done this way, as needed.

7) New application request - interactive, about 200 a year. This transaction logs in a few fields for new LEAs (such as the name, ID, address, contact person) and creates the new applicant request file throughout the year.

8) Initial allocation transaction - interactive, first allocation performed only once a year. Occasionally, this amount is amended, perhaps five times per year. This simply consists of editing the annual constants file.

9) Checkpoint transaction - interactively, the FOB and property sections can log on, edit the receipt control portion of the applicant and payment files, and check the flags there, indicating an OK for that application. These are performed once for every application per year. An identical transaction could easily be added for SAB, if desired.

10) Field report transaction - interactive, about 2,000 field reports are performed a year. This keeps information about the field report, as well as data generated for correcting applications. The data may be added at any pace or in any order, as desired.

The following three transactions (11, 12, 13) require special explanation because they are both the most common, and the most complex, of all the transactions. The complexity comes from the fact that these transactions perform multiple functions and they each require user intervention (and often input) before completion. Therefore, they are conceived to be in distinct parts, as follows:

- o The data base is searched, the needed records are retrieved, and the user prompting is completed if new information is needed,

- o An intermediate file, containing one or more update records, is produced,
- o The user must intervene by editing this intermediate file interactively and approving or rejecting records one by one, and
- o The intermediate file is applied to the data base files to complete the update for the permanent records.

All transactions must be completed, otherwise the data base will develop inconsistencies. Any of several mechanisms could be established to enforce this rule; perhaps the simplest would be that the existence of any of the transactions files indicates incomplete transactions, and they could be named or flagged in such a way as to force the users to complete them.

11) Update application data from field report or TR21 - This can be performed in an ad-hoc manner as desired, but the actions the program does will be mostly "canned," with user intervention at critical points. For field report updates, the program fetches the current applicant record, reads the field report log file data, and flags the appropriate receipt control fields. If data are to be changed based on the field report, the program will check child counts which must be reflected in property records, and prompt the user for these changes if necessary. If the applicant already received payment that year, the history record is written to the applicant and payment history file before update. If using a TR21 update, the same actions will be performed as needed. Therefore, both situations require that the user have property-specific data at hand before attempting these transactions; otherwise the program will not allow changes.

12) Initial payments transaction - Interactively started and controlled, but most of the program actions will be "canned." The program selects 200 - 300 applicants from the application and payment file, reads the annual parameters necessary, calculates the TR21 and TR22 data, calculates the year's obligation for those LEAs, and writes the obligation tape for FMS. It then calculates payments, generates reports on status, and creates the TR21 transaction file on disk. This TR21 temporary file can then be edited (or the report scanned) for final problems. Those stopped are deleted by hand, if desired, then the file is submitted for final payment action. At this juncture, the payment data are added to the application and payment file for the permanent record, the state and batch payment files are updated, and the payment disk pack is generated.

13) Follow-up payments transaction - This transaction is similar to the initial payments transaction, but applies to applicants after their obligation has been established and they have received payment. The program searches the queue of applicants fitting this description, distributes them geographically, and generates the temporary follow-up payments file. When this file is edited for problems, it is run against the main files, updating them. It also generates the payment disk, and updates state and batch payment files as required. In addition, the spread payment can be designed to be a variation of this transaction.

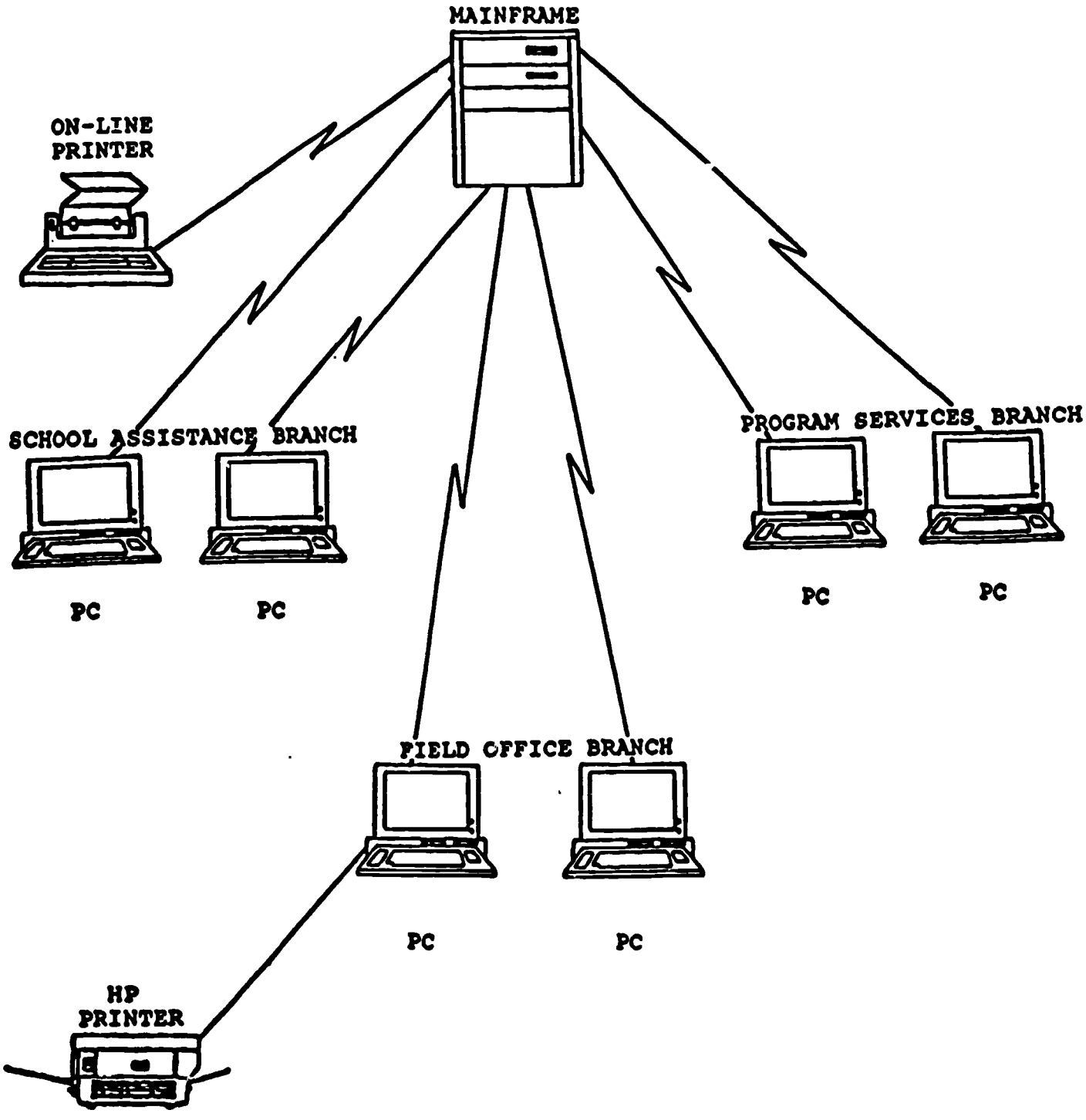
14) System purge and archive - This transaction is performed once a year, just before start-up processing for a new fiscal year. The transaction removes six-year-old data from the following files: application and payment, state level file, batch payment file, annual constants, application and payment history, and field report log file. The program would be run in batch and write data to tapes for permanent archives.

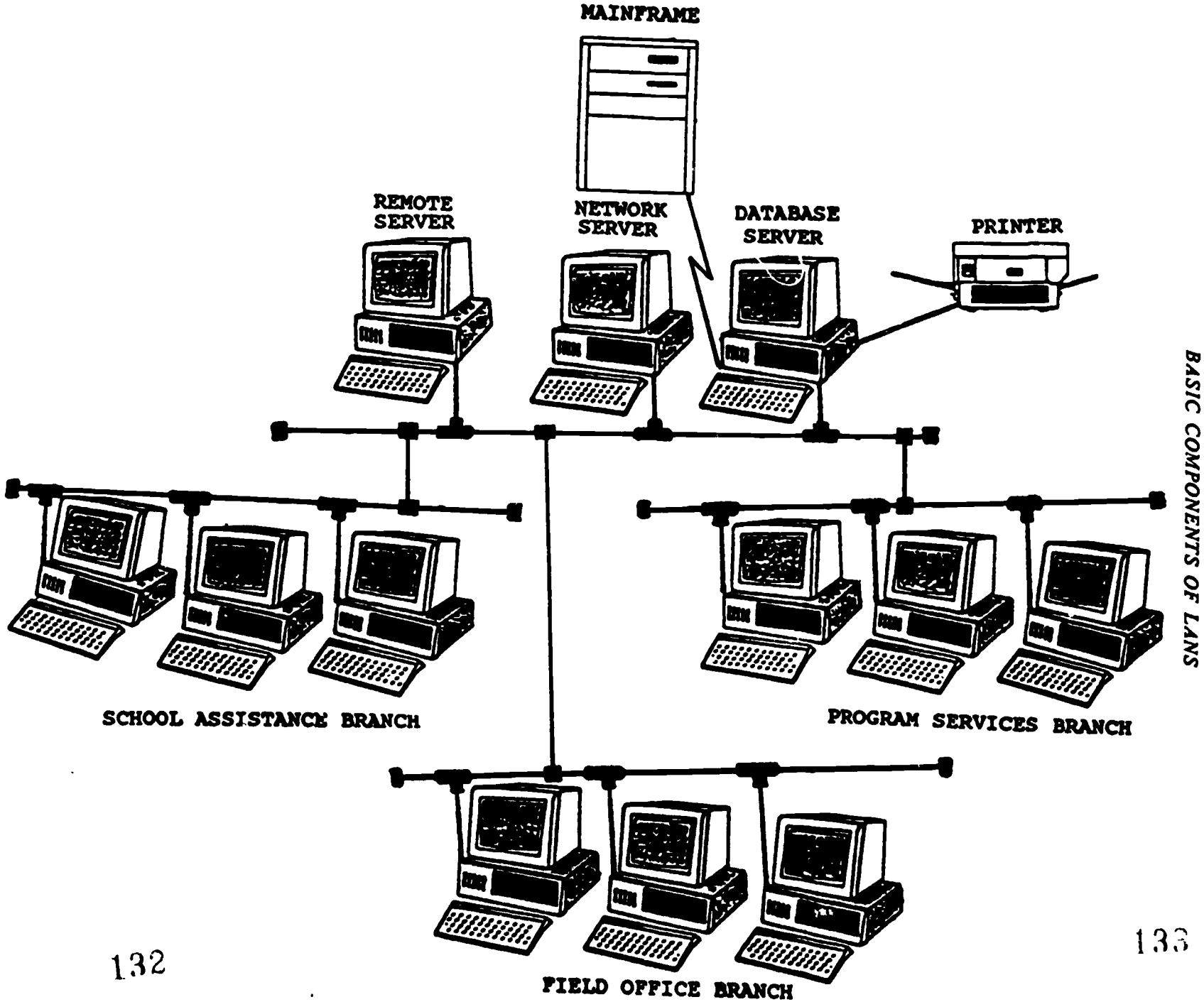
15) Preliminary application transaction - When a year's payments have begun, this transaction will be used to request preliminary payment for a continuing applicant. When the letter is received and approved, the user logs on and enters the ID. The program pulls up the previous year's payment record, computes the estimated payment (usually 75 percent of previous payment), and generates the temporary preliminary application file. This file is then used (after being scanned, if desired) to run against the main application files and generate payments, just as follow-up payments are done. The application and payment files are updated and a payment disk pack is produced.

16) Weekly backup transaction - This will be an entirely "canned" batch transaction generated by the DBA at least once per week. It reads the entire data base and writes it out to tapes, as a precaution against system failures.

ATTACHMENT M

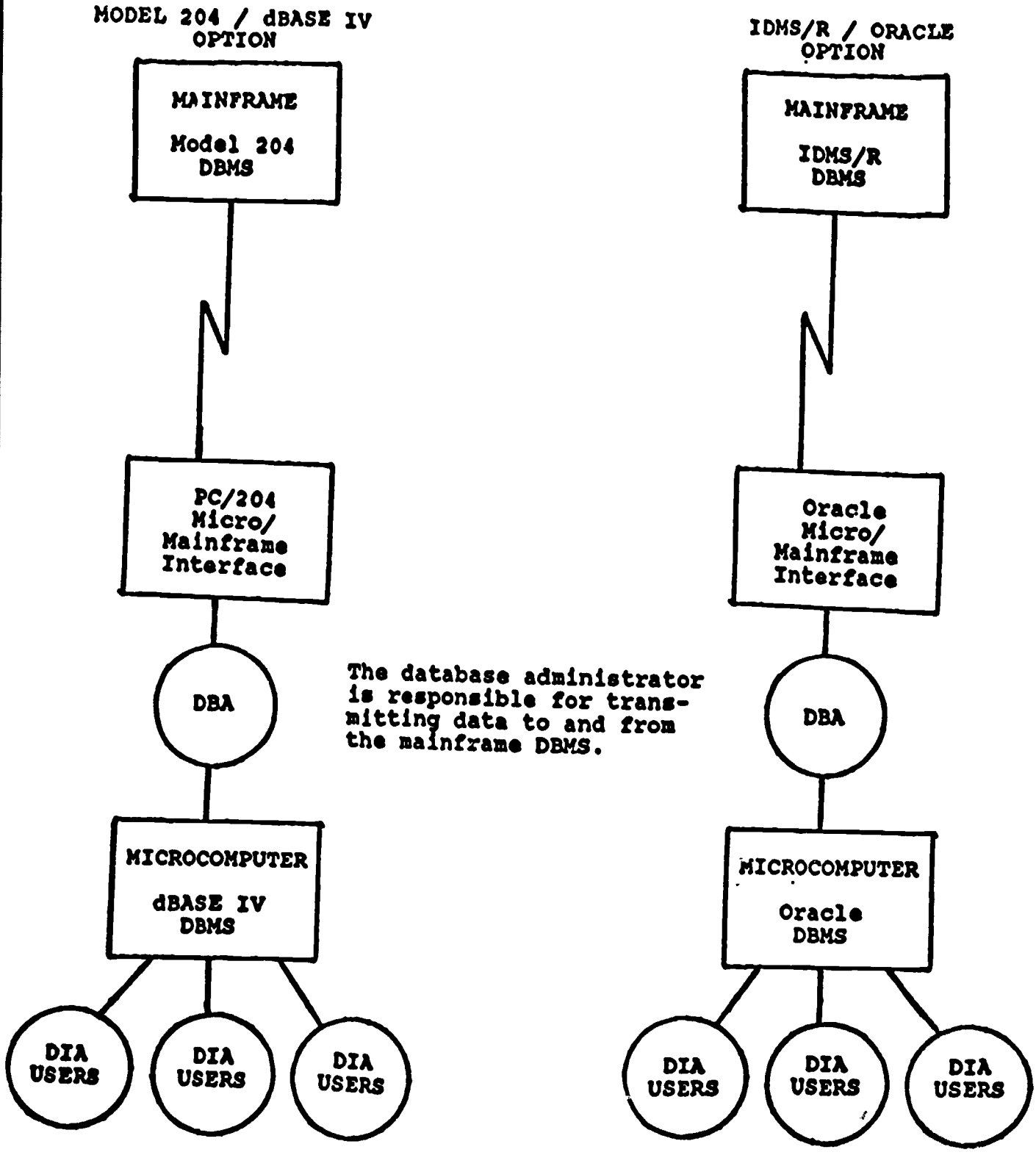
SCHEMATIC DIAGRAM OF THE MAINFRAME-ONLY SYSTEM




MAINFRAME
REMOTE SERVER
NETWORK SERVER
DATABASE SERVER
PRINTER
SCHOOL ASSISTANCE BRANCH
PROGRAM SERVICES BRANCH
FIELD OFFICE BRANCH

ATTACHMENT O

ILLUSTRATION OF SOFTWARE FOR THE MICRO-MAINFRAME COMBINATION SYSTEM



The database administrator is responsible for transmitting data to and from the mainframe DBMS.

DIA users will only be able to access data stored in the microcomputer DBMS, i.e., dBASE IV or Oracle, depending on the option selected. All mainframe access will be limited to the database administrator.

ATTACHMENT P

FILE IMPLEMENTATION IN A MAINFRAME/ MICROCOMPUTER SYSTEM

ID FILE

This file will continue to be maintained on the mainframe with an interactive menu system to be developed on the microcomputer for making changes to the data. This information will then be uploaded to the mainframe and run as a transaction file against the master listing. If there are no errors, the master file will be updated. The frequency of updating this file depends on the time of year. During the annual start-up phase, it may be necessary to update this file daily, while at other times it may be necessary to perform updates once or twice a week.

PROPERTY FILE

The property file, like the ID file, will continue to reside on the mainframe, and will be changed by PSB through an interactive menu system developed on the microcomputer. Once the new properties are entered, the data base administrator will upload this file to the mainframe, where it will be run as a transaction file against the master file. If no errors are found in the transaction file, the master file will be updated and the system will output a hard copy report of properties added, changed, or deleted from the master file. Like the ID file, the frequency of performing updates varies depending on the time of year.

APPLICATION AND PAYMENT FILE

Portions of the application and payment file will reside in both environments. The main component of all applications and the main components of Section 2 and Section 3 participants will reside on the microcomputer. These components contain control data which reflect the applicant's status, whether the applicant is a pre-approved LEA, entitlements under the various qualifying criteria, amount of payments, percent of entitlements, local contribution rates, and any amounts the LEA is to be debited. This file will need to be updated with every payment cycle made on the mainframe and downloaded to the microcomputer for quick availability.

BATCH PAYMENT DATA FILE

This file will be maintained in both environments. Batches of payments taking place on the mainframe will update this file, upon which the data will be downloaded from the mainframe. It is expected that this will be a weekly or bi-weekly process during the normal business cycle and performed several times a week in the beginning of the fiscal year.

ANNUAL STATE LEVEL FILE

This file will reside in both environments and be downloaded after every payment batch, so that the microcomputer always contains the most recent information. This file will contain LEA summary data by state and program section, such as amount of entitlements and payments made.

ANNUAL CONTROL DATA

This file is under the strict control of the SAB and will be maintained on the mainframe with access limited to the data base administrator. This file contains parameters necessary for payment processing, e.g., annual appropriation amounts, percent of entitlements, and current edit checks in force.

CONGRESSIONAL DISTRICT FILE

This file will be maintained only on the microcomputer. It is a master listing of Congressional districts, their representative, and address. Updating this file occurs as necessary and no special input menus are anticipated. This file is used for generating letters to representatives when a participating LEA in their jurisdiction receives their first payment under the program.

HISTORY OF APPLICATIONS AND PAYMENTS

This file will reside on the mainframe because of the amount of storage required and because it is not often needed. It is a record of all payment calculations for all LEAs participating in the Impact Aid program for the past five years. If, in the future, some of this data must be available for quick reference, portions of this file can be downloaded to the microcomputer for access across the network.

FIELD REPORT LOG FILE

This file will be accessible by the Field Office personnel responsible for keeping track of the status and resolution of site visits; it will reside on the microcomputer with portions of the file, i.e., the newly corrected data resulting from a field visit, uploaded to the mainframe for update processing. The frequency of updating this file depends on the time of year, volume of corrections or updates, and the importance of updating a particular LEA. Another possibility is that this file can be updated by Field Office personnel from a remote location, provided the Field Officer has access to a PC with a modem and telecommunications software. This has the advantage of speeding up the payments to an LEA which is under a field review.

CONTINUING APPLICANTS SHELL FILE

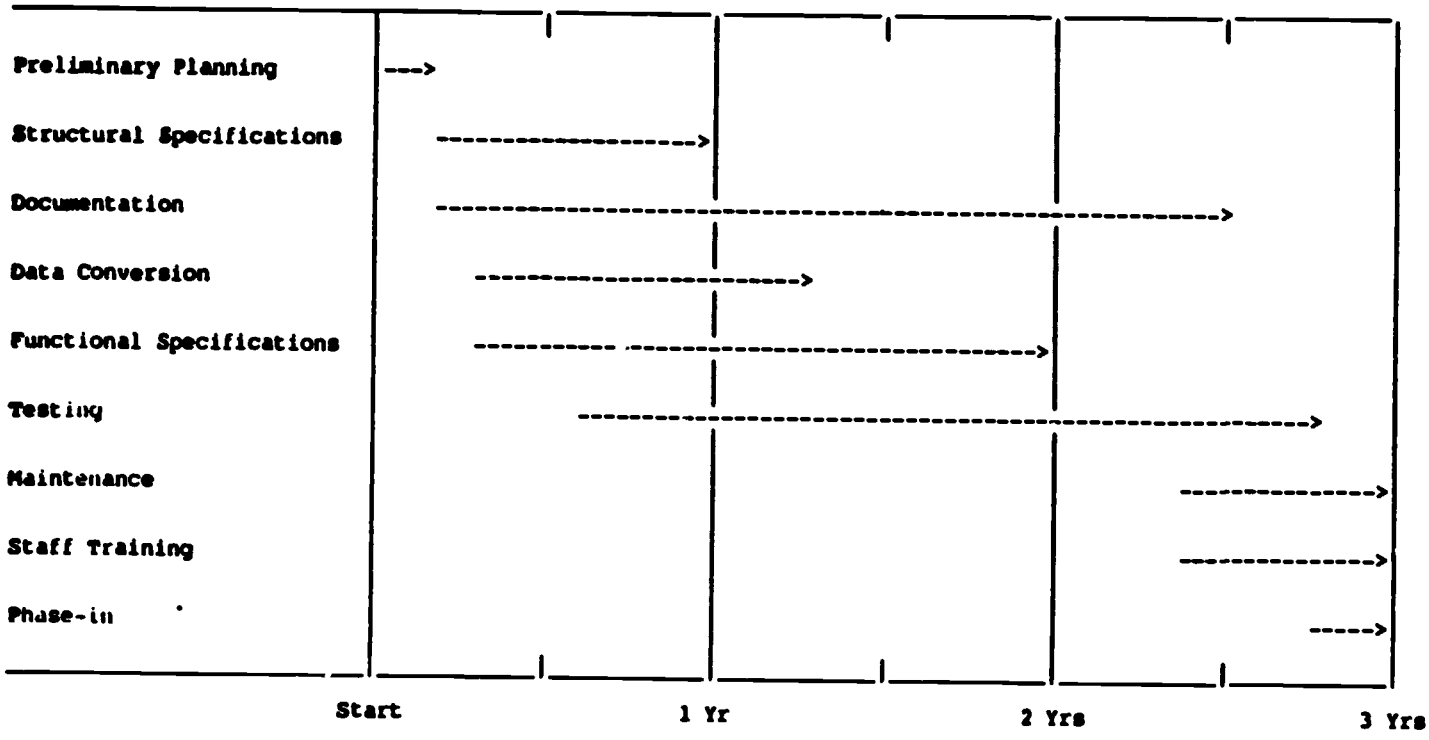
This file will reside on the microcomputer and contain data submitted by the applicant for the previous year. This file will be corrected and updated when the LEA submits the current year information. Once the data for the LEA are corrected, the application will be uploaded to the mainframe for payment processing and will be deleted from the microcomputer file. It is anticipated that this file will require daily uploading to the mainframe during the start-up phase when the applicants return their paperwork for the Impact Aid program.

NEW APPLICANT REQUEST LOG FILE

This file will reside on the microcomputer and contain information on new or non-continuing Impact Aid applicants. Data will be entered into this file through menu screens developed on the microcomputer. When new applications are keyed in each year, this file can be uploaded as a shell for the new applications, if desired.

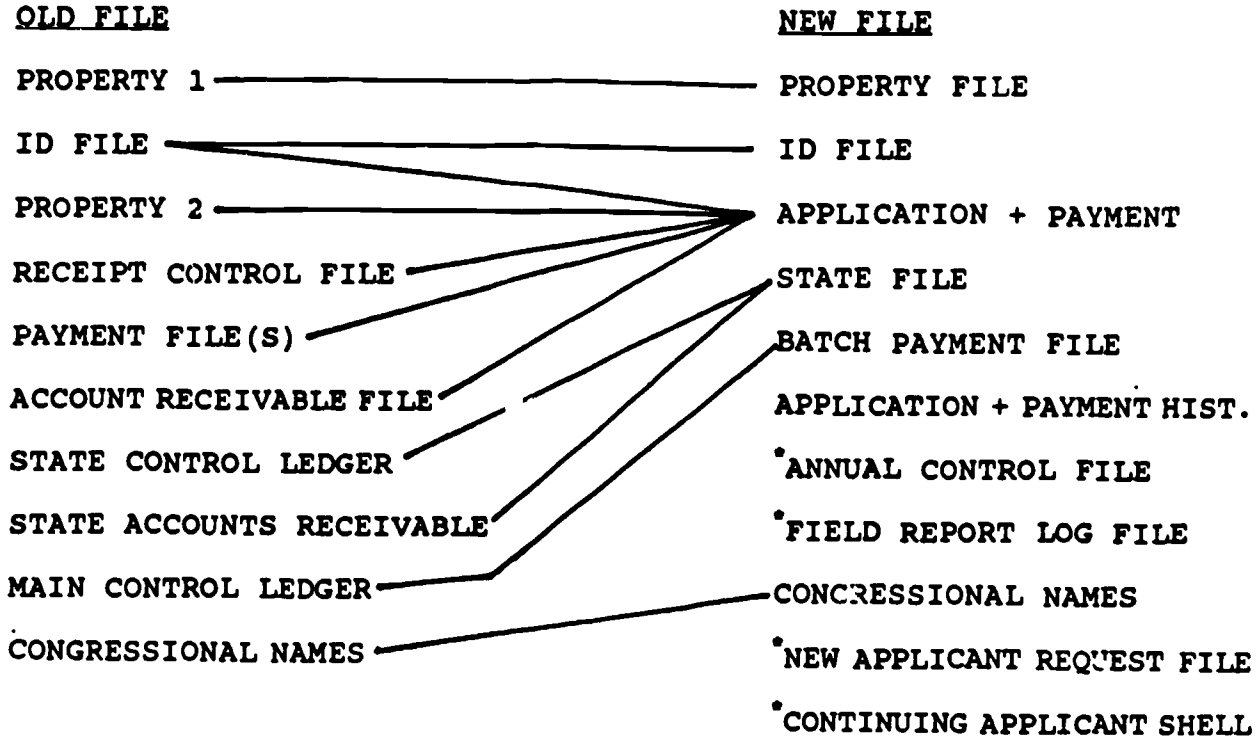
ATTACHMENT Q

TIMETABLE FOR IMPLEMENTATION OF OIA SYSTEM



ATTACHMENT R

MAPPING OF DATA FROM OLD FILES TO NEW, OIA DATA BASE



* indicates new file

GLOSSARY OF TERMS

ADP	Automated Data Processing (ED's computing center and its contractors)
ED	Department of Education
EIN	Employer Identification Number
FMS	Financial Management Service (in ED)
FOB	Field Office Branch (in OIA)
LEA	Local Education Agency
OIA	Office of Impact Aid
OIRM	Office of Information and Resource Management
OPBE	Office of Planning, Budget and Evaluation
PSB	Program Services Branch (in OIA)
SAB	School Assistance Branch (in OIA)
SAFA	School Assistance in Federally Affected Areas
SEA	State Education Agency