AUTOMATION OF DOMINION BANK’S
FEDERAL RESERVE BANK ACCOUNT

by

Shea Turman Chaney

Project submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTERS OF SCIENCE
in
System Engineering

APPROVED:

[Signature]
K. Triantis

[Signature]
B. Blanchard

[Signature]
J. Hicks

May, 1992
Blacksburg, Virginia
# CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Deficiencies of the Manual Process</td>
<td>2</td>
</tr>
<tr>
<td>Definition of Need</td>
<td>3</td>
</tr>
<tr>
<td>Feasibility Analysis</td>
<td>4</td>
</tr>
<tr>
<td>Functional Flow</td>
<td>4</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>6</td>
</tr>
<tr>
<td>Preliminary Design</td>
<td>12</td>
</tr>
<tr>
<td>Organization of the Report</td>
<td>13</td>
</tr>
<tr>
<td>2. BACKGROUND INFORMATION</td>
<td>15</td>
</tr>
<tr>
<td>Banking and Accounting</td>
<td>15</td>
</tr>
<tr>
<td>Dominion Bank</td>
<td>15</td>
</tr>
<tr>
<td>Banking and Accounting Terminology</td>
<td>15</td>
</tr>
<tr>
<td>Federal Reserve Bank</td>
<td>17</td>
</tr>
<tr>
<td>Federal Reserve Bank Internal Accounting System</td>
<td>17</td>
</tr>
<tr>
<td>General Ledger System</td>
<td>20</td>
</tr>
<tr>
<td>Initial State of Account 1060</td>
<td>22</td>
</tr>
<tr>
<td>Research Into Automation of Account Reconciliation</td>
<td>24</td>
</tr>
<tr>
<td>DISC System Versus PC System</td>
<td>25</td>
</tr>
<tr>
<td>Programming Language</td>
<td>25</td>
</tr>
<tr>
<td>3. DETAIL DESIGN</td>
<td>28</td>
</tr>
<tr>
<td>Short-Term Solution</td>
<td>28</td>
</tr>
<tr>
<td>Features</td>
<td>29</td>
</tr>
<tr>
<td>Reports</td>
<td>32</td>
</tr>
<tr>
<td>Long-Term Solution</td>
<td>36</td>
</tr>
<tr>
<td>GL Transaction Journal</td>
<td>37</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2-1</td>
<td>File Layout for FRB File</td>
</tr>
<tr>
<td>2-2</td>
<td>File Layout for GL File</td>
</tr>
<tr>
<td>3-1</td>
<td>File Layout for WDEBITS, WCREDITS, TDEBITS, TCREDITS</td>
</tr>
<tr>
<td>3-2</td>
<td>FRB Transaction Code Usage</td>
</tr>
<tr>
<td>3-3</td>
<td>Report Groups Within Balancing Group</td>
</tr>
<tr>
<td>3-4</td>
<td>File Layout for GL Data Base File</td>
</tr>
<tr>
<td>3-5</td>
<td>File Layout for FRB Data Base File</td>
</tr>
<tr>
<td>3-6</td>
<td>GL Requirements for Coin and Currency</td>
</tr>
<tr>
<td>3-7</td>
<td>GL Requirements for Automated Clearing House</td>
</tr>
<tr>
<td>3-8</td>
<td>GL Requirements for Cash Letters</td>
</tr>
<tr>
<td>3-9</td>
<td>GL Requirements for Food Stamps</td>
</tr>
<tr>
<td>3-10</td>
<td>GL Requirements for Bond Coupons</td>
</tr>
<tr>
<td>3-11</td>
<td>FTS File Layout</td>
</tr>
<tr>
<td>3-12</td>
<td>GL Requirements for AIM</td>
</tr>
<tr>
<td>3-13</td>
<td>GL Requirements for MBS and MBSPD</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-14</td>
<td>Relationship Between FRB Codes and MMII Identifiers</td>
</tr>
<tr>
<td>3-15</td>
<td>MMII File Layout</td>
</tr>
<tr>
<td>3-16</td>
<td>GL Requirements for MMII</td>
</tr>
<tr>
<td>3-17</td>
<td>GL Requirements for Trust</td>
</tr>
<tr>
<td>3-18</td>
<td>GL Requirements for Savings Bonds</td>
</tr>
<tr>
<td>3-19</td>
<td>GL Requirements for FRB Fees</td>
</tr>
<tr>
<td>3-20</td>
<td>GL Requirements for Correspondent Banks</td>
</tr>
</tbody>
</table>
# FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Functional Flow Diagram</td>
<td>7</td>
</tr>
<tr>
<td>3-1</td>
<td>Main Menu for Short-Term PC System</td>
<td>31</td>
</tr>
<tr>
<td>3-2</td>
<td>Report Menu for Short-Term PC System</td>
<td>33</td>
</tr>
<tr>
<td>3-3</td>
<td>Automated Matching System Menu Tree</td>
<td>104</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

In the Spring of 1990, Dominion Bankshares Corporation began a major corporate reorganization effort. The reorganization transformed the corporation from a series of regional affiliates to a line of business structure. With this reorganization came the consolidation of the bank’s five Virginia affiliates into a single Virginia bank. These changes dramatically impacted the work flow of the corporate accounting department located at the corporation’s headquarters in Roanoke, Virginia.

One accounting task in particular was so severely impacted by the consolidation that it eventually became unmanageable, posing a serious financial risk to the bank. That task was the reconciliation of the bank’s Federal Reserve Bank (FRB) account. The FRB account was being used by various areas throughout the corporation to facilitate a variety of financial transactions. Before the consolidation, each regional affiliate had been responsible for reconciling its own section of the account. Following the consolidation, the responsibility for all five regions was assigned to Corporate Accounting. There was no addition to the accounting staff and no training was provided to prepare for the new responsibilities; it remained assigned to the existing accounting staff, which included one clerk who was fully devoted to the FRB account, and two clerks who devoted half of their time to the FRB account.

The account reconciliation was being performed manually by Corporate Accounting when this project was initiated. At that time the account was being balanced only once every two weeks, and required almost ten hours per day were required to perform the manual process of matching all items on the bank
statements. No time was left for the research and resolution of items that could not be matched, and as a result the number of these exception items escalated rapidly.

As a systems analyst within the Financial Group, I was asked in January, 1991 to automate the FRB account. At that time, nearly 1300 unmatched items outstanding, totalling millions of dollars. This report describes the steps taken to automate the reconciliation of Dominion's FRB account and the SmartWare II database program that was developed to support a daily balancing capability.

DEFICIENCIES OF THE MANUAL PROCESS

Following the consolidation it became apparent that the manual reconciliation process was not adequate to support the increased volume of transactions in the FRB account. Generally speaking, little effort had been made to automate the tasks in Corporate Accounting. With the exception of a few spreadsheet projects, most accounting functions were performed manually. The corporate accounting department was very much behind the times with regard to system development and integration. The problems encountered with the FRB account are a direct result of the failure to recognize the need for the automation of accounting tasks and to invest in the personnel and equipment needed to achieve automation goals. The following are deficiencies of the manual reconciliation process:

- With the existing staff, the manual matching and balancing processes consumed more than ten hours daily, resulting in:
  - No resources available for researching and resolving exception items.
- The biweekly balancing requirement could not be met.
o The new requirement for daily balancing could not be met.
  o Only one person was familiar with the entire process; therefore continuation of process was entirely dependent on one person.
  o The process was not documented.
  o There were insufficient reports and paperwork to provide and audit trail.
  o The list of exception items was manually updated once every two weeks and was prone to error.
  o Balancing calculations were time consuming and prone to error.
  o There were no standards in place for reporting exception items back to the originator and no standards for resolving the items.
  o The GL entries were not standardized; as a result the descriptions being used were virtually meaningless.
  o Information on the status of the account was available only once every two weeks, when the balancing process occurred. Information on the status of the account was not kept current.
  o No steps had been taken to automate any function of the process on a PC or mainframe computer.
  o Out-of-balance conditions were difficult to trace because of the difficulty with calculations, often causing additional delays.

DEFINITION OF NEED

For the second half of 1991, executive management considered this project to have the highest priority of any project in the corporation. The FRB account is the most complex account in the bank’s ledger. This is due to many factors: the large number of transactions that pass through each day; the large number of areas in the bank that originate
these transactions; and the complexity of the transactions caused by reporting inconsistencies and delays.

In addition to the complex nature of the account, it is important because the Federal Reserve Board requires that a minimum cash reserve balance be maintained in the account at the close of each business day. This balance must be maintained as an offset to potential losses on assets. The amount that must be maintained in the account as cash reserves is specified by the Federal Reserve Board. Thus it is crucial that the bank verify the closing balance of this account on a daily basis. If too much cash is kept in reserves, the bank loses income on potential investments. If too little is kept in reserves then the bank could be fined by the Federal Reserve Board or could find itself unable to redeem liabilities.

Many of the transactions processed in this account involve large dollar amounts. Since the accounting staff did not have the resources available to research exception items, as time passed it became more unlikely that the bank could ever collect on these items. Therefore, the bank could have been sacrificing income on potential investments. Problems with account reconciliation are common throughout the banking industry. In a study conducted in June, 1991 it was found that 89% of banking institutions polled nationwide had a reconciling problem at least once during the last two years. [1] These problems were attributed to seven major causes (in order of decreasing occurrence): staff turn-over, no staff training, new software, volume of entries, mergers/acquisitions, operational flows, and software upgrades. [2]

**FEASIBILITY ANALYSIS**

Several alternative solutions to the problem were considered. The first alternative considered was the
continuation of the manual process. With the current staff, the requirements for daily balancing could not be met. Consequently at least two additional clerks would be needed. A considerable amount of time would be required to train the clerks and the cost for two full-time clerks would be approximately $40,000 annually with an expected increase of five percent annually. There was also a concern that five clerks working together on the same account would create the potential for even more calculation and matching errors due to the complexity of the process.

These concerns led us to consider a computer system to automate the process. Mainframe systems were considered; the favored candidate had an initial cost of $60,000 with support and maintenance costs over a five-year period totalling $1,000,000, and an implementation schedule of approximately eighteen months. This system was eventually purchased because it would permit the automation of accounts throughout the bank. Since the mainframe system would not be available for eighteen months the accounting managers insisted that a customized PC system be developed in-house to be used during the implementation period. The steps taken to standardize data for the PC system would simplify the implementation process for the mainframe system. Therefore the PC solution was chosen and pursued immediately, while the mainframe project was postponed for approximately four months.

**FUNCTIONAL FLOW**

Once it was determined that the manual process would be replaced with an automated process it was necessary to identify the functions of the system. The manual reconciliation process being used at the initiation of this project consisted of several steps that are described here in a simplified manner. Each day a report from Dominion's general ledger (GL) system,
listing transactions from the prior day, was compared to an account statement from the FRB. Similar items from the two reports were considered matches and were paired off and eliminated; any remaining items were considered to be exceptions. This matching process was performed daily. Once every two weeks the exception items were manually tabulated and the totals computed using a calculator. These totals were used in conjunction with the totals of matched items, and the current account balances, to balance the account. Figure 1-1 shows the functional flow envisioned for automated matching and balancing processes.

OPERATIONAL REQUIREMENTS

The long-term goal of the project was to eliminate the majority of the manual matching and data entry effort required to reconcile the account and to provide reporting capabilities to satisfy audit and accounting requirements. The specific operational requirements of the automated matching system are listed below.

Performance and Physical Parameters

- Provide a daily account balancing capability.
- System should attempt to automatically match current and exception items within a balancing group.
- User must be able to work interactively with system to download files daily and perform automated and manual matching for each balancing group.
- System must allow manual clearing of items.
- Control checks must be incorporated to ensure that no data is lost and that the account will not fall unintentionally out of balance.
- User must be able to select reports, and should be able to print or reprint them at any time during the day.
Figure 1-1
Functional Flow Diagram
Figure 1-1 (Con't)
Functional Flow Diagram
Figure 1-1 (Con't)
Functional Flow Diagram
Results must be mathematically error-free.

Processing time for automated matching should not exceed one hour each day. The remaining seven hours in the work day are needed for research and resolution of unmatched items. The following requirements are based on the manual process:

- Approximately 800 items must be compared daily
- Approximately 1300 items are currently outstanding.

Must be available for use in the Corporate Accounting Department building.

Program must be developed using the SmartWare II software Version 1.02. The data base files must be developed in the data base manager module and program must be developed in the SmartWare II project processing language.

Must operate on an IBM compatible PC, containing a 386 processor which provide the fastest processing available on departmental PC’s and will assure compatibility with other equipment.

Mainframe connectivity is required because source files must be downloaded from mainframe computer to the PC.

System must automatically back up critical files.

**Use Requirements**

- System must be available at all times.
- System will be in use approximately 8 hours each day.
- System should be menu driven, requiring minimal PC knowledge.

**Operational Deployment**

- System will be used on a Novell network, on several PC’s in the Corporate Accounting Department.
- System will become operational once a test with actual data is verified against manual reports and system can be installed on the user’s PC.
System documentation must be available to user.
- User documentation
- Report examples

Backup of files and hard copies of the program code should be stored off-site.
- One copy of program to be kept on programmer's PC internal hard drive.
- Additional copy of program to be kept on diskettes.
- Print out of all programs and file specifications to be kept by programmer.
- Report examples to be kept by programmer and by user.

Operational Life Cycle
- System will be used until a mainframe system can be installed - approximately 18 months.
- At conversion to new system, current inventory of unmatched items must be transferred to new system.

Effectiveness Factors
- Progress from a biweekly to a daily account reconcilement.
- Reduction in time required for reconcilement of account - from 10 hours per day to 8 or fewer hours per day with no additions to staff.
- Provides report capabilities currently unavailable.
  - Report of items automatically matched
  - Report of items manually cleared
  - Balancing sheet with outstanding items totals and account balance
  - Exception items listed by originating area, segregated by age of item
- Daily information available on status of account instead of biweekly.
- Reduction in mathematical errors - all calculations on
reports should be error-free.

- Faster resolution of unmatched items. The time limit on resolution of items will vary depending on where the item originated; limits vary from 5 to 30 days.

**Hardware and Operating Environment**

- System must run on IBM compatible PC with 386 or 486 processor.
- SmartWare II software and program files will run from PC internal hard drive (c:) or network file server.
- Backup of critical files to 3.5 inch diskettes.
- Reports to print on an Okidata 393 printer.
- Reports to print in 12-pitch, 8 lines per inch, on continuous feed 11 x 14 paper.
- PC must have access to mainframe computer.

**PRELIMINARY DESIGN**

The impetus for this project was the realization that a computer tool must be put in place immediately to achieve a timely reconciliation of the FRB account. The solution required two phases. The initial phase would be to develop a simple system to keep track of the accumulated exception items. This system would need to generate various reports containing the exception items and would eliminate the need to manually update the list. The use of automatically generated reports would eliminate calculation errors.

The long-term solution to the problem would be to create a system that could take the GL and FRB statements in a data format, match all possible items, and maintain the accumulated list of exception items. The system would require a much greater reporting flexibility to permit the reporting of exception items back to their originating areas for research and resolution.
The SmartWare II software package was the corporate standard for PC software when this project was initiated. The package included integrated modules for wordprocessing, data base, spreadsheet and PC communications. The spreadsheet and data base modules were the tools considered for development of the PC systems. I determined that the data base would best support the functional requirements of the short-term and long-term PC systems.

The computer system design and logic needed to be based on the manual reconciliation process. Thus the project began by studying each step of the manual process at a very detailed level. This included a study of the reports used in the matching process. Another element critical to the automation process was identifying the areas of the bank where transactions were being originated.

Automation of the matching process would be possible only if the data provided by the GL and FRB systems was similar enough to provide data that could be used as a basis for comparison. Thus the GL data needed to be modified to conform with the FRB reporting standards. The manual and automated entries to the GL system would need to be modified to conform with these standards. Therefore the originating areas would need to be informed of the changes required and their compliance would need to be monitored. The next step in the process would be to acquire the data files needed from the GL system, the FRB accounting system, and the other bank systems that were interfaced to the GL system. This data would need to be standardized for use in the automated matching program. The final step would be program development, testing and implementation.

**ORGANIZATION OF THE REPORT**

This report will cover background information needed to
understand the complexity and severity of the problem. In Chapter 2, terminology used throughout this report will be presented, followed by a discussion of related research and more specific background information on the project. Chapter 3 will cover in detail the development of the programs used as the short-term and long-term solutions. The results of the project will be discussed in Chapter 4, followed by conclusions and recommendations in Chapter 5.
CHAPTER 2
BACKGROUND INFORMATION
BANKING AND ACCOUNTING

Dominion Bank

Dominion Bankshares Corporation is a $10 billion bank holding company with its corporate headquarters in Roanoke, Virginia. The corporate reorganization, initiated in January, 1990, consolidated five regional Virginia banks (Southwestern Virginia, Shenandoah Valley, Greater Hampton Roads, Northern Virginia, Richmond) into one Virginia bank, which was then split into four lines of business (Corporate, Retail, Mortgage, Investments). The automation of the FRB account fell under the purview of the Corporate Accounting Department, which is organizationally under the Controller of the corporation.

Prior to the consolidation of the five Virginia banks, each region had its own accounting staff. Each regional accounting staff was responsible for reconciling its own piece of the FRB account, which often included correspondent banks. Correspondent banks are customers of Dominion and pay to have certain banking functions performed for them. Following the consolidation, the responsibility of all five regions and all correspondent banks fell to the corporate accounting staff, which was not prepared for the change.

Banking and Accounting Terminology

Certain banking and accounting terms are used throughout this paper and may be unfamiliar to technical readers. Such terms will be explained in this section. Most of these terms are taken from Banking Terminology: Third Edition published by the American Bankers Association (1989).

Account - A record of the financial transactions affecting the
assets, liabilities, income, expenditures, or net worth of an individual or business entity.

Audit Trail - A chronological record of activities that allows the reconstruction or review of the sequence of events for a transaction from its inception to final output.

Clearing Account - An intermediate account through which transactions are accumulated before being transferred to other accounts. Groups of costs, revenues, assets and liabilities are often moved through such accounts before being totalled and summarized elsewhere.

Credit - A transaction to an account representing positive cash flow. A credit to the FRB account would be a debit to the corresponding account on the general ledger. A "We Credit" is a credit to the GL account; a "They Credit" is a credit to the FRB account. For calculation purposes, credits are negative (-).

Debit - A transaction to an account representing negative cash flow. A debit to the FRB account would be a credit to the corresponding account on the general ledger. A "We Debit" is a debit to the GL account; a "They Debit" is a debit to the FRB account. For calculation purposes, debits are positive (+).

Exceptions - Transactions that fail to meet the parameters of the system and require special handling. Exceptions are usually displayed periodically on a listing called an exception report.

Interface - The tie-in between two data processing systems; a common boundary where automatic data-processing systems or parts of a single system meet and intersect.

General Ledger - A collection or book of accounts.

Reconciliation - A process of comparing and balancing one accounting record against another.

Timing Difference - The reconciling difference between a
ledger account balance and the corresponding subsidiary ledger balance created when two offsetting or related entries to the same ledger account fail to post in the same processing cycle. Once both entries clear the account or the processing cycles catch up, they offset and the timing difference is eliminated.

FEDERAL RESERVE BANK

The FRB is part of the Federal Reserve System which is the central banking authority of the United States. There are nine branches throughout the country; Roanoke is served by the Richmond, Virginia branch. The Board of Governors of the Federal Reserve System determines the reserve requirements of member banks, and the FRB's are the custodians of these accounts. In addition, FRB's also provide a variety of banking services to their member banks, such as cashing checks drawn on other member banks.

Federal Reserve Bank Internal Accounting System

The FRB has an Internal Accounting System (IAS) that is used to service the accounts of its member banks. The IAS brings together data from all of its systems that are involved in transactions with banks. Some examples of these systems are the Regional Delivery System which is used to process U.S. Savings Bonds sales, and Fed Line 2 which is a wire transfer system used to inform the FRB of a bank's transactions with other member banks.

The accumulation of all of a bank's transactions with the FRB are consolidated into a statement that is sent to each member bank on a daily basis. This statement reports each transaction that cleared through the FRB on the prior day and the opening and ending balances.

Before the initiation of this project, the statement was being printed each night at the FRB and then sent via
overnight mail to Dominion Bank along with several reports containing more detailed information for certain transactions. I explored the possibility of obtaining this information via electronic transmission because it would eliminate the delay caused by the mailing and because the information contained in the report would be needed in a data file for use in the automated matching system. I learned that it was possible to receive an electronic transmission of the report (including headers, footers and other cosmetic enhancements), and a separate file containing strictly data. Thus I initiated a request to the mainframe applications area to do the programming necessary to accommodate the transmission of two files from the FRB mainframe computer to Dominion's mainframe computer each night following the daily update cycle. The programming was accomplished within a matter of weeks and procedures were put in place for printing the account statement from the bank's mainframe computer. I began an analysis of the data file to determine how it could be used in the automated matching system.

The data file is made up of hundreds of records, each record being 200 characters in length. The file layout for the data file is contained in Appendix A. Each day's file begins with a header record containing the bank's opening and ending balances, and total immediate debits and credits. The header record is followed by detail records, each corresponding to a transaction found on the FRB statement. After analyzing the file layout of the data file, I determined that I would only need a subset of the available fields for the automated matching system. Therefore I had a mainframe programmer develop a job that would extract the needed fields from the data file and produce a modified data file for use in the PC system. The file that is used in the automated matching program is shown in Table 2-1.
### Table 2-1

File Layout for FRB File

<table>
<thead>
<tr>
<th>New Character Position</th>
<th>Original Character Position</th>
<th>Field Description</th>
<th>Length-Type</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>Statement Section</td>
<td>X(01)</td>
<td>I=Immediate, D=Deferred, F=Future</td>
</tr>
<tr>
<td>11-15</td>
<td>6-10</td>
<td>Transaction Code</td>
<td>X(05)</td>
<td></td>
</tr>
<tr>
<td>13-28</td>
<td>11-25</td>
<td>Amount</td>
<td>9(13).99</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>26</td>
<td>Debit/Credit</td>
<td>X(01)</td>
<td>C=Credit, D=Debit</td>
</tr>
<tr>
<td>1-9</td>
<td>27-35</td>
<td>Respondent Transit</td>
<td>X(09)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Routing Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29-32</td>
<td>67-70</td>
<td>Reference Number</td>
<td>X(04)</td>
<td></td>
</tr>
<tr>
<td>33-40</td>
<td>71-76</td>
<td>Process Date</td>
<td>(YYMMD)</td>
<td></td>
</tr>
<tr>
<td>41-48</td>
<td>77-82</td>
<td>Available Date</td>
<td>(YYMMD)</td>
<td></td>
</tr>
<tr>
<td>49-57</td>
<td>83-91</td>
<td>Control Item</td>
<td>X(09)</td>
<td></td>
</tr>
</tbody>
</table>
Below is an explanation of the use of each field:

**Statement Section** - Identifies each transaction as an immediate, deferred or future. Only the immediate items are considered for automatic matching.

**Transaction Code** - A five-character code used to segregate transactions by category. Only the first four characters are used consistently across FRB districts, the fifth character is for internal use in the Richmond FRB.

**Debit/Credit Indicator** - Indicates whether a transaction is a debit or credit to the FRB account.

**Respondent Transit Routing Number** - Identifies the nine-character ABA number of the bank that was involved in the transaction. This could be the ABA number of Dominion Bank or one of its correspondent banks.

**Reference Number** - A four-character field containing unique identifiers from other systems.

**Process Date, Available Date** - Date the transaction was processed or available at the FRB.

**Control Item** - Identifies unique data within a transaction code category.

Additional fields are used in the matching program, but they are derived from the data found in these fields.

---

**GENERAL LEDGER SYSTEM**

Dominion Bank maintains its general ledger of accounts on the Management Sciences America (MSA) General Ledger system. This system is maintained on the bank's mainframe computer at the operations center in Hollins, Virginia. Transactions may be entered for an account either manually or automatically. Manual entries are initiated when a preprinted ticket is completed with the necessary information about a transaction; the ticket is forwarded to the proof department where data entry personnel enter the information into a terminal
connected to the mainframe. GL tickets can originate in almost any area of the bank, including the branch offices, but the tickets are always sent to proof for entry into the GL system.

Automated entries originate from other computer systems in the bank. An example of this is the Money Market II system which is used to process security transactions for the broker/dealer subsidiary of Dominion Bank, which is Dominion Investment Banking, Inc. This system resides on a mainframe computer at the vendor’s location in Waltham, Massachusetts. Each night during the daily update cycle, a GL interface file is created by the system. It contains, in a summarized form, all of the GL transactions that occurred for that day. This file is transmitted to the bank’s mainframe computer, and then uploaded into the GL system. The automated interface takes the place of hundreds of manual entries that would need to be made. The summarization of entries provided by an automated interfaces speeds the nightly processing of the GL system, but discussed later in this paper, it makes the reconcilement of GL accounts much more difficult.

The FRB account is account number 1060. At various stages of this project, two other accounts were used to aid in the reconcilement process; account 1061 was used to isolate U.S. Savings Bonds transactions, and account 1036 is a clearing account used for deferred cash letter transactions. Following the nightly update of the GL system, numerous reports are produced, breaking down the information by cost center and account number. The log of transactions for a particular account is called the transaction journal. Each day the accounting department received a copy of the transaction journals for accounts 1060, 1036, and 1061. A copy of this report for account 1060 is contained in Appendix B. The transaction journal was needed in a data file for the automated matching program. Therefore I submitted a request
to the appropriate area of the bank requesting that the transaction journals for accounts 1060, 1036, and 1061 be combined in a data file to be produced daily. Knowing that the data in this file would need to provide a basis for comparison with the FRB file, I studied the information contained in the transaction journal and requested the file described in Table 2-2. This file was produced in ASCII format. Additional fields were needed for the automated matching program, but they are derived from the data provided in this file.

**INITIAL STATE OF ACCOUNT 1060**

At the initiation of this project, approximately three hundred transactions were passing through the FRB account daily. This number does not include the nearly five hundred wire transfer transactions passing through the account that were being interfaced to the GL as a net debit and credit. The number of exception items had grown to approximately 1300, which was approximately 300 in each of the four categories, "We Debits," "We Credits," "They Debits," and "They Credits." Some of these items were more than one year old. These items accounted for millions of dollars of unresearched outages and posed a serious threat to the bank.
Table 2-2

File Layout for GL File

<table>
<thead>
<tr>
<th>Character Position</th>
<th>Field Description</th>
<th>Length-Type</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>Company Number</td>
<td>X(04)</td>
<td></td>
</tr>
<tr>
<td>5-8</td>
<td>Account Number</td>
<td>X(04)</td>
<td></td>
</tr>
<tr>
<td>9-15</td>
<td>Center Number</td>
<td>X(07)</td>
<td></td>
</tr>
<tr>
<td>16-25</td>
<td>Source</td>
<td>X(10)</td>
<td></td>
</tr>
<tr>
<td>26-55</td>
<td>Description 1</td>
<td>X(30)</td>
<td>30 characters of descriptive text</td>
</tr>
<tr>
<td>56-67</td>
<td>Description 2</td>
<td>X(12)</td>
<td>12 characters of descriptive text</td>
</tr>
<tr>
<td>68-83</td>
<td>Amount</td>
<td>9(13).99</td>
<td></td>
</tr>
<tr>
<td>84-91</td>
<td>Effective Date</td>
<td>(MM/DD/YY)</td>
<td></td>
</tr>
<tr>
<td>92-99</td>
<td>Entry Date</td>
<td>(MM/DD/YY)</td>
<td></td>
</tr>
<tr>
<td>100-101</td>
<td>Debit/Credit</td>
<td>X(02)</td>
<td>10=Debit  60=Credit</td>
</tr>
</tbody>
</table>
RESEARCH INTO AUTOMATION OF ACCOUNT RECONCILIATION

It was determined early in the project that all of Dominion's peer banks were using some type of automated system for the reconciliation of their FRB account. Some banks had developed their own in-house systems, either for the PC or mainframe, while others had purchased systems. While it was not possible to obtain information on other banks' in-house systems, it was possible to evaluate purchased systems. Mainframe account reconciliation systems are available from the following vendors:

Recon-Plus  DISC, Inc., Baltimore, Maryland
Vector 8     Directions, Inc., Dallas, Texas
SOLV        Security Pacific Information Service, Denver, Colorado
SOAR        Microbank Software, New York, New York

DISC's Recon-Plus and Vector 8 were evaluated and the DISC system was eventually purchased. The mainframe systems provide extensive reporting capabilities and the ability to perform matching based on tolerances. Tolerance matching is the ability to match on a range of values for one or more of the matching criteria. For example, a pair of items could be matched if the transactions amounts fall within $5.00 of each other, as opposed to the exact match criteria built into the PC system. The mainframe systems also have the ability to perform one-to-many, and many-to-many matches, with verification by the user.

PC-based systems are offered by two vendors:

Universal Account Reconciliation  J. Driscoll & Associates, Dallas, Texas
Dimension 4000                     The Kirchman Corporation, Orlando, Florida

I evaluated the Universal Account Reconciliation system. This system did not have the ability to segregate transactions by group, nor the ability to use different matching criteria on
the basis of group. This system was found to be useful for very simple account reconciliations with matching based on unique transaction number and amount.

**DISC System Versus PC System**

The initial state of the FRB account was considered to be a financial threat. This situation was quickly communicated to the very highest levels of management. As a result, the need for better management of account reconcilement throughout the corporation became a high priority of management. While the FRB account required an immediate remedy, a long-term plan to automate and consolidate account reconcilement functions throughout the bank was initiated. A total of 109 general ledger accounts used by the Integrated Loan System, Demand Deposits, Savings and Certificate of Deposit systems were selected for automation. The DISC Recon-Plus system was purchased for this purpose and would eventually be used for automation of the FRB account. However, the complexity of the system and the resource requirements would delay its implementation for at least eighteen months. Therefore the PC-based automated matching program was pursued as an interim solution, to be used for a minimum of sixteen months.

**PROGRAMMING LANGUAGE**

The programming tool used for the development of the short-term and long-term PC programs is SmartWare II, from Informix Software, Inc., of Lexena, Kansas. This software was the corporation's standard software package for the duration of the project. It includes four modules: a word processor, a communications package, a spreadsheet and a data base manager. In addition, it provides a Program Development Language (PDL) that can be used to create programs utilizing the command functions available in each module.

Functionally, the short-term PC program needed to provide
the ability to add and delete items from a storehouse of exception items, and to provide a variety of reports of these exception items. Thus the system could have been developed in either a spreadsheet or a database. The database was chosen over the spreadsheet because of its versatility. The database manager offers very flexible reporting capabilities that can be used to print all records, or a subset of records, within a file. Records can be easily added or deleted using the data entry operating mode. The spreadsheet is more useful for performing complicated calculations between stationary cells and is not as easily used for the addition and deletion of items. Therefore the database module was implemented for both the short and long-term PC programs.

The following terms related to the database manager and the PDL are used throughout the discussion of the short and long-term systems. A basic understanding of these terms will be needed to follow the program listing for the long-term automated matching system which is included in the Programmer's Documentation in Appendix C. These definitions are based on various discussions from the software documentation, Smart Database, Informix Software, 1989.

**Database** - A collection of data organized around a specific need and arranged for easy retrieval.

**Database Management System** - A system that directs and controls all data that enters or leaves a user database.

**Database File** - A set of records composed of fields, each record representing a unique item.

**Database Field** - A set of characters, of designated length and type, containing a piece of information about a particular record.

**Calculated Field** - A field that is not entered by the user, but is calculated from an equation that may depend on the data in other fields of the same record.
Command/View Operating Mode - Allows the user to execute database commands by selecting key words and options from a menu.

Enter/Update Operating Mode - Allows the user to enter new records into a data base file or revise data previously entered into a record.

Data Base View - A screen, tied to one or more data base files, that allows the user to see and manipulate data within the file.

Append - Allows the user to attach data from one view to the end of the current view.

Index - A list of physical numbers of records containing desired data.

Key - A field or group of fields used to control the order of display of all of the records of a view, based upon field contents.

Load/Unload - Activates/deactivates a view.

Query - An expression that uses field contents to select or perform actions on certain records in a view.

Sort - Similar to a key, but can be used to order a subset of the records in a view.

Transact - A function that allows the contents of a certain field(s) in the current view to be sent to a similar field(s) in a related view based on field(s) that are similar between the two views.

Window - A portion of the screen containing a view. Window commands allow the user to manipulate windows to look at several files or different parts of the same file at the same time.
CHAPTER 3
DETAIL DESIGN
SHORT-TERM SOLUTION

The short-term solution to the FRB account reconciliation problem was to provide a storehouse for exception items that could accommodate the daily addition and deletion of items, and provide a variety of reports. This would eliminate the task of manually maintaining the list of over 1300 exception items and transcribing the list and calculating exception item totals on a biweekly basis.

The manual matching process was performed most efficiently by dividing the transactions into groups of related transactions. On the FRB statement, transactions were segregated by a five-character transaction code which was automatically assigned by the IAS system based on the source of the transaction. Over 500 codes had been defined by the FRB and approximately half of them appeared on the statement on a daily basis. Some examples of these codes and their meanings are shown below:

05030 - Deferred Credits
08040 - Food Coupons
08110 - Treasury Call
10010 - Transfer of Funds
10020 - Transfer of Funds Reversed
10191 - Net Settlement
15010 - Consolidated City Cash Letters
15163 - Deferred Cash Letters

Over the years, the accounting staff had been able to attach meaning to the vague description associated with each code.

The identification of transactions appearing on the GL transaction journals was a more difficult task. Three fields, the Source Code, Description 1 and Description 2, should have given sufficient information to identify the source of the transaction. Unfortunately the Description 1 and Description
2 fields were not standardized in their usage and the information provided in these fields was essentially meaningless for the majority of the transactions entered on a daily basis. Therefore the clerk often relied on the Source Code field and the amount field to match transactions with the FRB statement.

Over time it was determined that dividing the transactions, on both sides, into ten distinct groups was the most efficient way of balancing the account because each of the ten groups could be balanced independently. All of the transactions reported by the FRB could be assigned to one of these groups based on the transaction code. Assigning the GL transactions to these groups was much more difficult and thus a more refined breakdown of the transactions could not be achieved consistently. The ten balancing groups are listed below with the group identifier used in the PC system:

- CB - Correspondent Banks
- ACH - Automated Clearing House
- CC - Coin and Currency orders
- CL - Cash Letters
- CPN - Bond coupon collections
- FS - Food Stamp collections
- FTS - Funds Transfer System (wire activity)
- INV - Investments
- OTH - Fees and unidentified transactions
- SB - U.S. Savings Bonds

Each of these groups will be discussed in detail later in this report because these groupings are the foundation for the automated matching system.

Features

The basis of the short-term PC system was a set of four data base files: "We Debits," "We Credits," "They Debits," and "They Credits" (file names WDEBITS, WCREDITS, TDEBITS, TCREDITS). Exception items were entered into a file based on
their origination, GL ("we") or FRB ("they"), and the type (debit or credit). From the main menu, shown below, the user could select the desired file, which would come to the screen in the data enter/update mode, and the user could add new items or mark existing items for deletion, which required the user to enter a date in the CLEAR_DATE field. When a file was unloaded, a query was executed that would delete any records with a valid date in this field. Deleted records were stored in a temporary file for use in a report. Table 3-1 shows the file layout for the four data base files.
GL ACCOUNT 1060 RECONCILIATION

A) CLEAR/ADD WE DEBITS
B) PRINT WE DEBITS SUMMARY
C) CLEAR/ADD WE CREDITS
D) PRINT WE CREDITS SUMMARY
E) CLEAR/ADD FRB DEBITS
F) PRINT FRB DEBITS SUMMARY
G) CLEAR/ADD FRB CREDITS
H) PRINT FRB CREDITS SUMMARY
I) REPORTS MENU
X) EXIT

Choice:

Figure 3-1

Main Menu for Short-Term PC System

Table 3-1

File Layout for WDEBITS,WCREDSITS,TDEBITS,TCREDSITS

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>JADDED</td>
<td>Alpha</td>
<td>1</td>
</tr>
<tr>
<td>GROUP</td>
<td>Alpha</td>
<td>5</td>
</tr>
<tr>
<td>ENTRY_DATE</td>
<td>Date(MM/DD/YY)</td>
<td>8</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Alpha</td>
<td>25</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>Numeric</td>
<td>15</td>
</tr>
<tr>
<td>CLEAR_DATE</td>
<td>Date(MM/DD/YY)</td>
<td>8</td>
</tr>
</tbody>
</table>
The use of each field is described below:

**JADDED** - an "A" would be entered automatically for new items. New items were stored in a temporary file for use in a report.

**GROUP** - the group code for each item (CB, ACH, CC, CL, CPN, FS, FTS, INV, OTH, SB).

**ENTRY_DATE** - the date the item appeared on the GL transaction journal or FRB statement.

**DESCRIPTION** - a meaningful description of the item.

**AMOUNT** - transaction amount.

**CLEAR_DATE** - date the item was matched and subsequently deleted from the file.

The main menu provided the user with the ability to add or clear exception items and to print a summary report of each file. Each file was automatically copied to a diskette each time the file was unloaded, and also when the user exited the main menu. This backup feature ensured that the most current files could be restored if a hardware failure destroyed the data on the PC hard disk.

**Reports**

A variety of reports were made available to the user. The report menu could be accessed from the main menu and provided the following options:
GL ACCOUNT 1060 RECONCILIATION

A) DETAILED REPORT BY GROUP
B) MBS ITEMS BY TRADE NUMBER
C) ADDED/CLEARED ITEMS
D) STALE ITEMS MENU
E) STALE ITEMS OVER 90 DAYS MENU
F) CLEAR/ADD FRB CREDITS
G) DAILY ACTIVITY COVER SHEET
H) RETURN TO MAIN MENU
X) EXIT

Choice:

Figure 3-2

Report Menu for Short-Term PC System
The contents of the reports are discussed below:

**DETAILED REPORT BY GROUP** — Within each of the ten balancing groups, potential matches were placed side by side. "We Debits" were printed next to "They Credits," and "We Credits" were printed next to "They Debits" in ascending amount order. This report allowed the accounting staff to compare the exception items within a group for possible matches. The programming for this report became quite complex because of the report limitations of SmartWare II. The software would not print records from multiple files on the same report. Therefore, the program was designed to send the report output to text files, which were then read into a double-column document in the word processor.

**MBS ITEMS BY TRADE NUMBER** — Within the INV group (investments), was a subset of transactions related to MBS. These transactions originated in the Funds Management area of the bank and involved the trading of MBS securities with other banks and brokers. By the nature of the trades, many securities would be traded together at one time. The FRB reported the transaction amount for each security involved in the trade, while the Funds Management area would report the trade total across all securities. Thus the only way to relate the FRB transactions to the GL transactions was to use the Funds Management-assigned trade number. Thus the four character trade number was entered in the first four characters of the DESCRIPTION field for each side of the trade. This report would select only MBS-related items from the four files, consolidate items from the INV group for all four files, and then print a report sorting the items by trade number. The MBS trades were found to be very complex, owing to timing delays in the reporting of the transactions, and a special report was a necessity.

**ADDED/CLEARED ITEMS** — This report provided a listing of new
and cleared transactions. When new items were added to any of the four files, an "A" was automatically entered into the JADDED field. When the file was unloaded, a copy of these items would be stored in a temporary file. The same held true for any items that were cleared by entering a date into the CLEAR_DATE field. This report provided the audit trail necessary to trace additions and deletions to the data base files.

**STALE ITEMS MENU** - A series of reports was made available for reporting exception items that had been outstanding for more than a designated number of days. The staleness criteria differed for each group. A separate menu gave the user the option of printing stale items within a specific group or for all groups. The report showed the age of the items in days and sorted the items in increasing age order within a group. These reports were sent to the originating areas for resolution.

**STALE ITEMS OVER 90 DAYS MENU** - A series of reports was made available for reporting items more than 90 days old. A separate menu gave the user the option of printing these items for a specific group or for all groups. These reports were sorted by increasing age within a group. These reports were sent to the executive managers of the originating areas for resolution.

**DAILY ACTIVITY COVER SHEET** - This report provided a summary of the daily activity in the account and permitted the staff to balance the account. Each day the user would enter the total debit and credit activity reported in the FRB statement and on the GL transaction journal. The program would calculate the total amounts of items outstanding. If the account were in balance, the difference between the current day's exception item total, and the prior day's exception item total should equal the difference in net activity in the account reported
by the FRB and GL. If these amounts were not equal, then the account was out of balance. This report was very useful in identifying out of balance conditions.

The short-term PC system provided an easy-to-use tool for maintaining the growing list of exception items. The reporting features of the system eliminated the need for manual calculations and simplified the balancing process. Within a few months of its implementation, it permitted a transition from a biweekly balancing to daily balancing. This was a very significant step in the process of automating the account reconcilement process. It not only added to its accuracy, but also forced the managers to initiate procedures and standards for the reporting of stale and very stale items which would eventually lead to resolution of all items more than thirty days old.

**LONG-TERM SOLUTION**

The first step towards the automation of the reconcilement of the FRB account was to achieve an in-depth understanding of the manual process, and to document the process at a detailed level. This was necessary not only for the automation project, but to ensure that continuation of the process would not be dependent on a specific person. The manual process involved the comparison of items between the FRB statement and the GL transaction journal. However, this process was complicated because the GL transactions were not reported in a manner that lent itself to comparison with the FRB statement. In addition many GL transactions were reported in a summarized form that required use of supplementary reports containing the transaction detail. Therefore the manual process involved the comparison of many reports, not just the FRB statement and the GL transaction journal.
GL Transaction Journal

The GL transaction journals for accounts 1036, 1061 and 1060 were delivered daily to the Corporate Accounting department. An example report for account 1060 is contained in Appendix B. These reports provided a listing of the transactions that are posted to each account the previous day. The Description 1 field, thirty characters in length, was intended to be used by the originating areas for a meaningful description of the transaction. The Description 2 field provided an additional twelve characters for any additional information. It is evident from the transaction journal example that for most transactions the descriptions provided little or no meaningful information to aid in their identification.

In some cases, the Source Code field was useful in identifying a transaction. A code beginning with "1V" indicated that the transaction originated from the Western Virginia Operations Center (WVOC); a "2V" indicated origination from the Eastern Virginia Operations Center (EVOC); "35010003" indicated an FTS interfaced entry, and "7502000002" a MMII interfaced entry. However entries to account 1060 were made by locations as remote as branch offices, therefore the Source Code provided little help in most cases.

The origination of the transactions within each balancing group will be presented in detail later in this chapter, but it is important to note that some of the originating areas provided supplemental reports for their GL entries.

- FTS - an interface summary and detailed listing of transactions
- Cash Letters (CL) - a report detailing immediate and deferred cash letter amounts
- MBS (INV) - transaction listing by security within a
trade and paydown amounts by security
  o MMII (INV) - listing of transactions that make up
    interfaced amount, by customer holding

FRB Statement
The daily FRB statement consists of approximately twenty
pages of transactions. An example of the statement is
contained in Appendix A. The statement is segregated by bank,
which separates Dominion Bank from its correspondent banks.
Within a bank, transactions are segregated by FRB transaction
code, which are five-character numeric codes. Within some
transaction codes, transactions are further identified by a
code in the four-character Reference field, nine-character
Control Item field, or nine-character Offset DPI field. In
addition to the statement, several other reports are received
daily:
  o MBS, MBSPD, MMII, AIM, TRUST (INV) - An Advice of Credit
    report containing supplementary information on the
    transactions reported in codes 20011, 20012, 20111,
    20191, 20201, 27011, 27021 (which are all
    security-related codes);
  o FTS - A report listing the individual transactions
    which make up the total debit and credit reported for
    code 10011, Transfer Funds. The debit and credit amounts
    reported in this code are the daily totals for hundreds
    of wire transfer transactions.
  o CL - Advice of Credit for cash letters
  o FS, SB - Advice of Adjustments report for food stamps
    and U.S. Savings Bonds
  o SB - Bond sales report from the Regional Delivery System
  o OTH - Summary statement of service charges (received
    monthly)
At the start of this project, all of the FRB reports were printed at the FRB and then shipped via overnight mail to the bank. Knowing that I would need the statement information in a data file, I arranged to have the statement (and data file) electronically transmitted to the bank’s mainframe each night, where it was printed and sent to the accounting department using the internal mail system. The supplementary reports were not made available for electronic transmission until December, 1991.

Tieing FRB Transaction Codes to Originating Areas

Even a cursory review of the FRB statement and GL transaction journal, makes it apparent that the reports did not lend themselves to a match and pair-off process because there was little similarity between them. While the FRB reported transactions in specifically defined categories, the information in the bank’s GL system was not standardized. This lack of standardization occurred not only in the use of descriptions, but in some cases the GL contained summarized entries where the FRB reported detail. The process could not be automated unless the data sets provided some foundation for comparison. It took a considerable amount of effort to convince management that this function could not be automated if it could not be performed manually.

Since the FRB report format was standardized nationwide, Dominion Bank had no choice but to adopt the same reporting standards. The first step towards automation was to determine which of the more than four-hundred FRB transaction codes were being used on a regular basis, and to understand the types of transactions that fell within these categories. Once we understood the usage of each FRB code, we could develop a set of guidelines for each originating area to use when making entries to account 1060. Successful implementation of the
standards on our GL system would simplify the manual matching process and provide the foundation needed for the automation process. The implementation of new standards would affect numerous areas of the bank, down to the branch office level, and would affect not only manual entries, but entries made by interfacing computer systems.

The first four characters of the FRB transaction codes are used consistently throughout all Federal Reserve districts; the fifth character is used at the discretion of the individual FRB banks for further refinement of the transaction categories. Therefore this fifth character would not be used in our reporting standards. The first two characters of the code provide a general breakdown of transactions, while the second two characters provide a more refined breakdown. Table 3-2 shows the usage of the first two characters.
Table 3-2
FRB Transaction Code Usage

<table>
<thead>
<tr>
<th>Char 1 &amp; 2</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>Matured Debits &amp; Credits</td>
</tr>
<tr>
<td>08</td>
<td>Treasury or Government Agency Service</td>
</tr>
<tr>
<td>10</td>
<td>Transfer of Funds</td>
</tr>
<tr>
<td>15</td>
<td>Cash Letters</td>
</tr>
<tr>
<td>20</td>
<td>Transfer of Securities</td>
</tr>
<tr>
<td>27</td>
<td>Principal/Interest of Government Agency Securities</td>
</tr>
<tr>
<td>30</td>
<td>Return Letter Items</td>
</tr>
<tr>
<td>40</td>
<td>Collection</td>
</tr>
<tr>
<td>50</td>
<td>Inter Office Transactions</td>
</tr>
<tr>
<td>57</td>
<td>Settlement Between Offices for ACH/EFTS Transactions</td>
</tr>
<tr>
<td>63</td>
<td>Coin &amp; Currency Transactions</td>
</tr>
<tr>
<td>66</td>
<td>Account Service or Stock Activity</td>
</tr>
<tr>
<td>70</td>
<td>Savings Bonds</td>
</tr>
<tr>
<td>82</td>
<td>Loans</td>
</tr>
<tr>
<td>84</td>
<td>Account Charges</td>
</tr>
</tbody>
</table>
Two months were spent studying the usage of the FRB transaction codes and two vital tasks were accomplished. Having achieved a detailed understanding of the FRB codes, it was possible to develop a set of standards for their usage by the originating areas. The standards specify the usage of the codes within the ten balancing groups. The changes requested of the originating areas differed slightly for each balancing group, but one change was required of all groups: the appropriate FRB code would be required in the first four characters of the Description 1 field. The implementation of these standards was initiated by sending a memorandum to each originating area, which came to a total of more than four-hundred mailings. A description of the specific changes required for each balancing group will later in this chapter.

It was also determined that many transactions could be identified to a level of detail beyond balancing group; therefore, most balancing groups were further refined. Balancing would still be done at the balancing group level, but the more detailed groupings would permit greater reporting specificity. Table 3-3 lists the report groups contained within each balancing group.
<table>
<thead>
<tr>
<th>Balancing Group</th>
<th>Report Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td>CBACH</td>
<td>Correspondent Bank - ACH</td>
</tr>
<tr>
<td></td>
<td>CBCL</td>
<td>Correspondent Bank - Cash Letters</td>
</tr>
<tr>
<td></td>
<td>CBOTH</td>
<td>Correspondent Bank - Other</td>
</tr>
<tr>
<td>ACH</td>
<td>EACH</td>
<td>Eastern VA Op Center ACH</td>
</tr>
<tr>
<td></td>
<td>WACH</td>
<td>Western VA Op Center ACH</td>
</tr>
<tr>
<td>CC</td>
<td>CCGH</td>
<td>Coin &amp; Currency Greater Hampton Roads</td>
</tr>
<tr>
<td></td>
<td>CCNA</td>
<td>Coin &amp; Currency Southwest Virginia</td>
</tr>
<tr>
<td></td>
<td>CCNV</td>
<td>Coin &amp; Currency Northern Virginia</td>
</tr>
<tr>
<td></td>
<td>CCR</td>
<td>Coin &amp; Currency Richmond</td>
</tr>
<tr>
<td></td>
<td>CCSV</td>
<td>Coin &amp; Currency Shenandoah Valley</td>
</tr>
<tr>
<td>CL</td>
<td>ECL</td>
<td>EVOC Cash Letters</td>
</tr>
<tr>
<td></td>
<td>WCL</td>
<td>WVOC Cash Letters</td>
</tr>
<tr>
<td></td>
<td>ERET</td>
<td>EVOC Returns</td>
</tr>
<tr>
<td></td>
<td>WRET</td>
<td>WVOC Returns</td>
</tr>
<tr>
<td>CPN</td>
<td>WVCPN</td>
<td>WVOC Coupons</td>
</tr>
<tr>
<td></td>
<td>TRCPN</td>
<td>Trust Coupons</td>
</tr>
<tr>
<td>FS</td>
<td>EVOC</td>
<td>EVOC Food Stamps</td>
</tr>
<tr>
<td></td>
<td>WVOC</td>
<td>WVOC Food Stamps</td>
</tr>
<tr>
<td>INV</td>
<td>MM</td>
<td>Money Market II</td>
</tr>
<tr>
<td></td>
<td>AIM</td>
<td>Aim Portfolio</td>
</tr>
<tr>
<td></td>
<td>MBS</td>
<td>MBS Trades</td>
</tr>
<tr>
<td></td>
<td>MBSPD</td>
<td>MBS Paydowns</td>
</tr>
<tr>
<td></td>
<td>TRUST</td>
<td>Dominion Trust Company</td>
</tr>
<tr>
<td></td>
<td>OTH</td>
<td>Unidentified Trust Investments</td>
</tr>
</tbody>
</table>
Considerations for Automation of Source Files

The information contained in the FRB statement and the GL transaction journals formed the basis for the manual matching process. This information was obtained through data files from the IAS and GL systems respectively. Tables 2-1 and 2-2 contain the file layouts for these files. However the data provided in these files needed to be transformed for usage in the automated matching program. The data base manager software contains a relate function that can be used to compare records in two different files using one or more fields as the basis for the comparison. The GL and FRB files were to be compared on the basis of balancing group code, debit/credit indicator, and amount (and for some balancing groups an additional field was also used for comparison). Thus the fields that would be used for comparison required standardization which was made possible by the use of calculated fields. Tables 3-4 and 3-5 show the file layouts for the GL and FRB data base files into which the GL and FRB data files are imported daily. An explanation of the use of each field follows each table.
### Table 3-4

File Layout for GL Data Base File

<table>
<thead>
<tr>
<th>Field</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Data Input</td>
<td>Type</td>
<td>Status</td>
</tr>
<tr>
<td>Format</td>
<td>Mask</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPANY</td>
<td>gl</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>ACCOUNT</td>
<td>gl</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>CENTER</td>
<td>gl</td>
<td>A 7</td>
<td>R/W</td>
</tr>
<tr>
<td>SOURCE</td>
<td>gl</td>
<td>A 10</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC_1</td>
<td>gl</td>
<td>A 30</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC_2</td>
<td>gl</td>
<td>A 12</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>gl</td>
<td>N 8</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Key: AMOUNT

Len: 8  Type: A  Dupes: Yes

| EFF_DATE     | gl              | D 8         | R/W          |

Key: EFF_DATE

Len: 8  Type: A  Dupes: Yes

| ENTRY_DATE   | gl              | D 8         | R/W          |
| DEBCRE       | gl              | A 2         | R/W          |
| DESC1        | gl              | A 4         | R/W          |

Calc: left([DESC_1],4)

| DESC1T       | gl              | A 26        | R/W          |

Calc: right([DESC_1],26)

| DESC2        | gl              | A 12        | R/W          |

Calc: [DESC_2]

| FILE         | gl              | A 3         | R/W          |

Calc: "GL"

| WORDS        | gl              | A 6         | R/W          |

Calc: if [DEBCRE]="10" then "DEBIT"
else "CREDIT"

| SIGN         | gl              | N 8         | R/W          |

Calc: if [DEBCRE]="10" then [AMOUNT]
else -1*[AMOUNT]

| GROUP        | gl              | A 5         | R/W          |

Key: GROUP

Len: 5  Type: A  Dupes: Yes

Calc: if
(right(left([DESC1T],5),4)="0283" or right(left([DESC1T],5),4)="0376" or right(left([DESC1T],5),4)="0416" or right(left([DESC1T],5),4)="0460" or right(left([DESC1T],5),4)="7255" or right(left([DESC1T],5),4)="7308" or right(left([DESC1T],5),4)="8288" or right(left([DESC1T],5),4)="8135" or right(left([DESC1T],5),4)="8194" or right(left([DESC1T],5),4)="8342" or right(left([DESC1T],5),4)="8199") then "CB"
else if ([DESC1]="5701" or [DESC1]="5703" or [DESC1]="5704")
<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-field Table Input</th>
<th>Data Length Type</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>then &quot;ACH&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>else if ([DESC1]=&quot;6301&quot; or [DESC1]=&quot;6304&quot; or [DESC1]=&quot;6305&quot;)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>then &quot;CC&quot;</td>
<td>else if ([DESC1]=&quot;0503&quot; or [DESC1]=&quot;1019&quot; or [DESC1]=&quot;1501&quot; or [DESC1]=&quot;1502&quot; or [DESC1]=&quot;1515&quot; or [DESC1]=&quot;1528&quot; or [DESC1]=&quot;1531&quot; or [DESC1]=&quot;1533&quot; or [DESC1]=&quot;3005&quot; or [DESC1]=&quot;3008&quot; or [DESC1]=&quot;3010&quot; or [DESC1]=&quot;3016&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>then &quot;CL&quot;</td>
<td>else if ([DESC1]=&quot;4001&quot;) then &quot;CPN&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>then &quot;FS&quot;</td>
<td>else if ([DESC1]=&quot;0804&quot; or [DESC1]=&quot;0806&quot; or [DESC1]=&quot;0808&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>else if ([DESC1]=&quot;1001&quot; or [DESC1]=&quot;1002&quot;) then &quot;FTS&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>else if ([DESC1]=&quot;2001&quot; or [DESC1]=&quot;2004&quot; or [DESC1]=&quot;2011&quot; or [DESC1]=&quot;2019&quot; or [DESC1]=&quot;2020&quot; or [DESC1]=&quot;2701&quot; or [DESC1]=&quot;2702&quot; or [DESC1]=&quot;2704&quot; or [DESC1]=&quot;4007&quot;)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>then &quot;INV&quot;</td>
<td>else if ([DESC1]=&quot;0802&quot; or [DESC1]=&quot;0811&quot; or [DESC1]=&quot;0812&quot; or [DESC1]=&quot;0813&quot; or [DESC1]=&quot;8403&quot; or [DESC1]=&quot;8418&quot; or [DESC1]=&quot;8415&quot; or [DESC1]=&quot;8422&quot; or [DESC1]=&quot;8425&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>then &quot;OTHER&quot;</td>
<td>else if ([DESC1]=&quot;7001&quot; or [DESC1]=&quot;7004&quot;) then &quot;SB&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>else if ([DESC1]=&quot;FTS &quot;) then &quot;FTS&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>else &quot;??????&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RGROUP
Calc: if [GROUP] = "CB" and
{[DESC1]="0503" or [DESC1]="1019" or [DESC1]="1501" or [DESC1]="1502" or [DESC1]="1515" or [DESC1]="1528" or [DESC1]="1531" or [DESC1]="1533" or [DESC1]="3005" or [DESC1]="3008" or [DESC1]="3010" or [DESC1]="3016") then
"CBCL"

else if [GROUP]="CB" and
{[DESC1]="5701" or [DESC1]="5703" or [DESC1]="5704") then
"CBACH"

else if [GROUP]="CB" and
{[DESC1]="0503" and [DESC1]="1019" and [DESC1]="1501" and [DESC1]="1502" and [DESC1]="1515" and [DESC1]="1528" and
<table>
<thead>
<tr>
<th>Field Display Format</th>
<th>Data-file Table Input Mask</th>
<th>Data Length</th>
<th>Entry Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>[DESC1]&lt;&gt;&quot;1531&quot; and [DESC1]&lt;&gt;&quot;1533&quot; and [DESC1]&lt;&gt;&quot;3005&quot; and [DESC1]&lt;&gt;&quot;3008&quot; and [DESC1]&lt;&gt;&quot;3010&quot; and [DESC1]&lt;&gt;&quot;3016&quot; and [DESC1]&lt;&gt;&quot;5701&quot; and [DESC1]&lt;&gt;&quot;5703&quot; and [DESC1]&lt;&gt;&quot;5704&quot;</td>
<td>else null</td>
<td>gl</td>
<td>A</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 3-4 (Con’t)

File Layout for GL Data Base File

Field Usage

COMPANY, ACCOUNT, CENTER, SOURCE, DESC_1, DESC_2, AMOUNT, EFF_DATE,
ENTRY_DATE, DEBCRE - imported from the data file

DESC1 - first four characters of DESC_1, which should contain the FRB code

DESC1T - 26 remaining characters of DESC_1 field. This allows preservation of the original entry description.

DESC2 - contents of DESC_2

FILE - represents the source of the item ("GL")

WORDS - turns the debit/credit indicator (DEBCRE) into "DEBIT" or "CREDIT" for use in reports

SIGN - attaches a sign to the amount so that net activity can be calculated (credits are negative (-) and debits are positive (+))

GROUP - the balancing group code which is determined by the contents of DESC1. If an invalid code is entered, the item is classified as unidentified and must be assigned a balancing group code by the user.

RGROUP - the report group code which must be entered by the user

MATCH - matching code. For automated matches this code is an "A" followed by the month and day the match occurred (mmdd). For manually cleared items, this code is made up of the user initials followed by a sequential number.
## Table 3-5

File Layout for FRB Data Base File

<table>
<thead>
<tr>
<th>Field</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>FRB Mask</td>
<td>A 9</td>
<td>R/W</td>
</tr>
<tr>
<td>ABA</td>
<td>FRB</td>
<td>A 1</td>
<td>R/W</td>
</tr>
<tr>
<td>STMT_SEC</td>
<td>FRB</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>TRANS_CODE</td>
<td>FRB</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Key:</strong> TRANS_CODE</td>
<td></td>
<td>Len: 5</td>
<td>Type: A</td>
</tr>
<tr>
<td>DC_CODE</td>
<td>FRB</td>
<td>A 1</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>FRB</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>REF</td>
<td>FRB</td>
<td>A 8</td>
<td>R/W</td>
</tr>
<tr>
<td>PRO_DATE</td>
<td>FRB</td>
<td>D 9</td>
<td>R/W</td>
</tr>
<tr>
<td>AVA_DATE</td>
<td>FRB</td>
<td>D 9</td>
<td>R/W</td>
</tr>
<tr>
<td>CON_ITEM</td>
<td>FRB</td>
<td>A 2</td>
<td>R/W</td>
</tr>
<tr>
<td>BANK_DFI</td>
<td>FRB</td>
<td>A 2</td>
<td>R/W</td>
</tr>
<tr>
<td>DESCRE</td>
<td>FRB</td>
<td>A 29</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Calc: if [DC_CODE]="D" then "60" else "10"

DESC1  
Calc: left([TRANS_CODE],4)

DESC1T  
Calc: if [ABA]="051402835" then "0283 BANK OF FLOYD"
else if [ABA]="051403766" then "0376 GRAYSON NATL"
else if [ABA]="051404163" then "0416 LEE BANK & TRUST"
else if [ABA]="051404600" then "0460 TAZEWELL"
else if [ABA]="251472555" then "7255 BEDFORD"
else if [ABA]="251473088" then "7308 SOUTHWEST VA"
else if [ABA]="251482888" then "8288 CARILION"
else if [ABA]="251481355" then "8135 NEWPORT NEWS"
else if [ABA]="251481944" then "8194 NAVAL NEWS"
else if [ABA]="251483421" then "8342 YORKTOWN"
else if [ABA]="251481999" then "8199 PFD FIREFIGHTERS"
else [DESC1T]

DESC2  
Calc: if [DESC1]="1001" then [BANK_DFI] else [REF]

FILE  
Calc: "FRB"
<table>
<thead>
<tr>
<th>Field</th>
<th>Display</th>
<th>Data-file Table</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Input</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td></td>
<td>FRB</td>
<td>A</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Calc: if [DC_CODE]=&quot;D&quot; then &quot;DEBIT&quot; else &quot;CREDIT&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td></td>
<td>FRB</td>
<td>N</td>
<td>8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Calc: if [DC_CODE]=&quot;D&quot; then [AMOUNT] else -1*[AMOUNT]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td></td>
<td>FRB</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>Key:GROUP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BANK_DFI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calc: if ([ABA] = "051402835" or [ABA]="051403766" or [ABA]="051404163" or [ABA]="051404600" or [ABA]="251472555" or [ABA]="251473088" or [ABA]="251482888" or [ABA]="251481355" or [ABA]="251483421" or [ABA]="251481999") then "CB"

else if left([DESC1],2)="57" then "ACH"

else if left([DESC1],2)="63" then "CC"

else if ([DESC1]="1011" or [DESC1]="1012" or [DESC1]="1019" or [DESC1]="1020" or [DESC1]="0502" or [DESC1]="0503" or left([DESC1],2)="15" or left([DESC1],2)="30") then "CL"

else if ([DESC1]="4001" or [DESC1]="4002") then "CPK"

else if ([DESC1]="0804" or [DESC1]="0806" or [DESC1]="0808") then "FS"

else if ([DESC1]="1001" or [DESC2]="1002" or [DESC1]="1099") then "FTS"

else if ([DESC1]="4003" or [DESC1]="4007" or [DESC1]="4009" or left([DESC1],2)="20" or left([DESC1],2)="27" or left([DESC1],2)="66" or left([DESC1],2)="82") then "INV"

else if left([DESC1],2)="70" then "SE"

else if ([DESC1]="0000" or [DESC1]="05C1" or [DESC1]="0504" or [DESC1]="4004" or [DESC1]="4005" or [DESC1]="4006" or [DESC1]="4008" or [DESC1]="4009" or [DESC1]="0505" or left([DESC1],2)="08" or
### Table 3-5 (Con't)

**File Layout for FRB Data Base File**

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Format</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input</td>
<td>Type</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```plaintext
left([DESC1],2)="50" or left([DESC1],2)="84" )
then "OTHER"
else "?????"
```

<table>
<thead>
<tr>
<th>RGROUP</th>
<th>FRB</th>
<th>A</th>
<th>5</th>
<th>R/W</th>
</tr>
</thead>
</table>
| Calc: if [GROUP] = "CB" then if ([DESC1]="1011" or [DESC1]="1012" or [DESC1]="1019" or [DESC1]="1020" or [DESC1]="1011" or [DESC1]="0502" or [DESC1]="0503" or left([DESC1],2)="15" or left([DESC1],2)="30") then "CBCL"
else if [GROUP]="CB" and (left([DESC1],2)="57") then "CBACH"
else "CBOTH"
else if [GROUP]="CC" then if ([ABA]="05100141" or [ABA]="051401674" or [ABA]="051403892") then "CCR"
else if ([ABA]="051400662" or [ABA]="051401328" or [ABA]="051403999") then "CCGH"
else if ([ABA]="051401409" or [ABA]="051401564" or [ABA]="051403520") then "CCSV"
else if ([ABA]="054001220" or [ABA]="055003201" or [ABA]="056004393") then "CCNV"
else "CCNA"
else if [GROUP]="SB" then if ([ABA]="05100141" or [ABA]="051401674" or [ABA]="051403892") then "SBR"
else if ([ABA]="051400662" or [ABA]="051401328" or [ABA]="051403999") then "SBGH"
else if ([ABA]="051401409" or [ABA]="051401564" or [ABA]="051403520") then "SBSV"
else if ([ABA]="054001220" or [ABA]="055003201" or [ABA]="056004393") then "SBNV"
else "SBNA"
else if [GROUP]="ACH" then if ([ABA]="05100141" or [ABA]="051400662" or [ABA]="051401328" or [ABA]="051401564" or [ABA]="051403520" or [ABA]="051403999" or [ABA]="054001220" or [ABA]="055003201" or [ABA]="056004393") then "EACH"
else "WACH"
else if [GROUP]="CL" then
Table 3-5 (Con’t)

File Layout for FRB Data Base File

<table>
<thead>
<tr>
<th>Field Display Format</th>
<th>Data-file Table Input</th>
<th>Data Length</th>
<th>Entry Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>if ([ABA]=&quot;051001414&quot; or [ABA]=&quot;051400662&quot; or [ABA]=&quot;051401328&quot; or [ABA]=&quot;051401409&quot; or [ABA]=&quot;051401564&quot; or [ABA]=&quot;051401674&quot; or [ABA]=&quot;051403520&quot; or [ABA]=&quot;051403892&quot; or [ABA]=&quot;051403999&quot; or [ABA]=&quot;054001220&quot; or [ABA]=&quot;055003201&quot; or [ABA]=&quot;056004393&quot; or [ABA]=&quot;056007604&quot;) then &quot;ECL&quot; else &quot;WCL&quot; else if [GROUP]=&quot;FTS&quot; then &quot;FTS&quot; else if [GROUP]=&quot;OTHER&quot; then &quot;OTH&quot; else null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MATCH</td>
<td>FRB</td>
<td>A</td>
<td>5</td>
</tr>
</tbody>
</table>
**Field Usage**

ABA, STMT_SEC, TRANS_CODE, DC_CODE, AMOUNT, REF, PRO_DATE, AVA_DATE,
CON_ITEM, BANK_DFI -- imported from the data file

DESCRE - indicates a debit or credit on the GL (an FRB debit is a GL credit and is coded as "60", an FRB credit is a GL debit and is coded as "10")

DESC1 - first four characters of the TRANS_CODE field, which is the FRB transaction code

DESC1T - for correspondent bank items this field indicates the bank name is used for reporting purposes

DESC2 - used to store data used within certain balancing groups as a basis for matching

FILE - represents the source of the item ("FRB")

WORDS - turns the debit/credit indicator (DC_CODE) into "DEBIT" or "CREDIT" for use in reports

SIGN - attaches a sign to the amount so that net activity can be calculated (credits are negative (-) and debits are positive (+))

GROUP - the balancing group code which is determined by the contents of DESC1. If an invalid code is entered, the item is classified as unidentified and must be assigned a balancing group code by the user.

RGROUP - the report group code. For some items it can be determined by the FRB transaction code and the ABA field

MATCH - matching code. For automated matches this code is an "A" followed by the month and day the match occurred (mmdd). For manually cleared items, this code is made up of the user initials followed by a sequential number.
Considerations for Automation of Each Bank Area

The ten balancing groups are used to divide each day's activity into groups of similar transactions. The matching and reporting functions of the PC program are based on these groups. The factors that were considered for the automation of each group are discussed below.

CC - Coin and Currency

The regional FRB's supply coin and currency to bank branches throughout the country; therefore the coin and currency orders placed by Dominion Bank branch offices pass through the FRB account. The branch offices place orders for coin and currency shipments using one of three methods: the bank's wire transfer system (Funds Transfer System (FTS)); placing a voice order via telephone; or utilizing a private armored carrier service. The following list identifies how orders are placed by the five Virginia regions:

- Southwest Virginia (DBNA) - For most branch offices orders are placed by the Money Department at the corporate headquarters using FTS. Those branches that do not utilize the Money Department place their own orders using FTS.

- Richmond (DBR) - All orders are placed by the branch offices using FTS.

- Greater Hampton Roads (DBGHR) - Coin orders are placed by voice order. Currency shipments are provided by Brinks Armored Carrier Service which do not pass through the FRB account.

- Northern Virginia (DBNV) - All orders are placed by the branch offices using FTS.

- Shenandoah Valley (DBSV) - All orders are placed by the branch offices using voice order.

If a branch office places its own orders, it is responsible for making the appropriate entry to the GL. The Money
Department makes the appropriate GL entry if it places the order for a branch office. The manner in which these entries were being made was inconsistent. Sometimes a branch would combine the coin and currency amounts for a branch into one entry. In other cases, the Money Department would combine entries for coin and currency together, or would combine orders for different branch offices together. The use of the Description 1 field was inconsistent and did not identify the source of the entry (e.g. "CASH SHIPPED 41/49," "BASSETT COINS SHIPPED," "COIN SHIPPED BLFD").

The FRB statement reports coin shipments under code 6301 - Coin Transaction, by individual branch office. Currency shipments are reported in code 6304 - Currency Transaction, also by individual branch office. Adjustments to any of these transactions are reported under code 6305 - Currency Adjustment. The FRB also reports the sequence number it assigns to each order in the Reference field of each item. The Fed-assigned sequence number is visible on the FTS transaction screen, and is reported to the ordering party verbally when voice orders are placed. The automation of this piece of the account required that the branch offices conform to the FRB reporting standards for coin and currency orders as shown in Table 3-6.

The modification of the manual entries to GL permitted automatic matching coin and currency items using the FRB code, transaction amount and sequence number as the matching criteria. Most adjustment items still require manual matching. Coin and currency exception items are reported back to each affiliate which is facilitated by the use of the report groups CCGH, CCNA, CCNV, CCR, and CCSV.
Table 3-6

GL Requirements for Coin and Currency

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
</table>
| 6301     | Description 1 Field (for coin transaction)  
|          | o FRB transaction code              
|          | o Transaction description - "COIN"   
|          | o Branch Name                       | Description 2 Field  
|          | o 4-digit sequence number of shipment |
| 6304     | Description 1 Field (for currency transaction)  
|          | o FRB transaction code              
|          | o Transaction description - "CURRENCY" |
|          | o Branch name                       | Description 2 Field  
|          | o 4-digit sequence number of shipment |
| 6305     | Description 1 Field (for coin or currency adjustment)  
|          | o FRB transaction code              
|          | o Transaction description - "COIN" or "CURR" "ADJ" |
|          | o Branch name                       | Description 2 Field  
|          | o For shipments to FRB, 4-digit sequence number of shipment | o For shipments from FRB, last 4 digits of 16-digit branch identification number |
ACH - Automated Clearing House

Dominion Bank serves as the clearing house for southwest Virginia banks. A clearing house facilitates the exchange of local checks and the settlement of balances arising out of that exchange. Each member receives credit in the clearing settlement for the total amount of checks it presents drawn on other participant banks, and it is debited or charged with the total amount of checks drawn on it and presented by the other banks. The payment of the net settlement amounts is accomplished through the FRB; the net debit and credit balances are posted to the member bank's accounts at the FRB.[3]

ACH activity is processed in the Electronic Transfer Processing (ETP) departments at WVOC and EVOC. The GL entries are made using batch entry sheets to reflect the processing of transactions for all Dominion Bank affiliates. Prior to the standardization of GL entries for ACH activity, affiliate activity was often reflected as another affiliate's item. The situation was corrected by swapping items between the affiliates. This swap activity resulted in differences between the GL posted activity and the FRB statement. The situation was further complicated by ACH items that were returned as not collectable. The Description 1 field of the GL entries typically contained a 4-character bank code (which corresponds to characters 5-8 of the 9-character bank ABA number) followed by "ACH DEBITS" or "ACH CREDITS" (e.g. "7255 ACH CREDITS"). While this description permitted the items to be identified as belonging to the ACH group, it did not differentiate between original transactions, adjustments, and swaps.

The FRB statement reports ACH items by affiliate under FRB codes 5701 - Inter-Zone ACH Items, and 5703 - ACH-Return Items. Thus the ETP departments needed to conform with the
FRB reporting conventions for ACH transactions. The following changes were requested for ACH entries.
Table 3-7
GL Requirements for Automated Clearing House

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>5701</td>
<td>Description 1 Field (for ACH transactions and adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (&quot;ETP&quot;)</td>
</tr>
<tr>
<td>5701</td>
<td>Description 1 Field (for ETP batch settlement sheet entries)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o &quot;ETP SETTLEMENT&quot; or &quot;ETP FED RETURNS&quot;</td>
</tr>
<tr>
<td></td>
<td>o Center Location</td>
</tr>
<tr>
<td>5701</td>
<td>Description 1 Field (for all ACH swap transactions)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Department Name - &quot;ETP&quot;</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Date of swap</td>
</tr>
</tbody>
</table>
These changes permitted automatic matching of most original and return transactions using the FRB code and transaction amount as the matching criteria. However, since the swaps are generated internally they must be matched manually. ACH exception items are reported to the ETP departments at EVOC and WVOC which is facilitated by the use of report groups ECH and WACH.

**CL - Cash Letters**

Cash letters are groups of checks that must all go to the same bank for payment. All checks cashed or deposited at Dominion Bank are sorted by drawee bank and are bundled together and submitted to the FRB who acts as an intermediate paying agent. At Dominion, cash letters are processed on the CPCS system which is used at two locations, WVOC and EVOC. Cash letters for which immediate credit of the full amount is expected are totalled each day by affiliate and reported to the GL. There are some cash letters whose paying agent is out of the Richmond FRB district. For these cash letters the FRB may have to wait several days before it receives payment, therefore the credits to our account are staggered over a several days according to a predetermined disbursement scheme. The total amount of cash letters for which we expect to receive deferred credit is posted to account 1036, which is a clearing account. Through a complex allocation scheme, this money is moved into account 1060 as our FRB account is credited. The CPCS system provides reports listing the individual cash letters that constitute the immediate and deferred totals and these reports are sent to the accounting department each day.

The matching process for cash letters is complicated by the fact that there are many return items. Return items are cash letters that are rejected by the FRB because they have
incorrect amounts, items listed which were not enclosed, or contain coding errors. These cash letters are called "exceptions" or "returns" and can be incoming or outgoing. The FRB corrects such errors and reports these items with a reference to the original cash letter amount.

The FRB statement has a very detailed breakdown of codes for reporting cash letter activity. All codes beginning with "15" (of which there are 45) report original cash letter transactions within a single FRB district. The most common codes are:

1515 - Intra City Cash Letter (immediate)
1523 - Postal Money Order (immediate)
1524 - Government Checks (immediate)
1528 - Cash Letter Adjustments
1540 - Savings Bond Cash Letter (immediate)

All codes beginning with "30" (of which there are 15) report cash letter returns. The most common code in this category is 3005 - Local Return Cash Letter. Exceptions for return cash letters are reported in code 5704. Code 0503 - Deferred Informational Credits is used to report the total deferred amount credited daily. The cash letter items are reported in total by affiliate, with the exception of 0503 which reports one total for all regions. The individual cash letter amounts that make up the totals reported in the statement are reported on the Cash Letter Advice of Credit report. In this report the original transaction amount is referenced for return items.

Automation of the cash letter group was very complex because of the deferred/immediate pay schedule, and the large number of adjustment items. In addition, neither the GL nor the IAS system provided the cash letter detail needed for matching individual cash letter transactions. I explored the possibility of obtaining the needed detail information from the CPCS system for which there was two alternatives. A data file containing the needed detail could have been generated
daily and downloaded from the mainframe computer into the PC system; or an automated interface to the GL could have been created which would send the detail directly to the GL system. Several meetings were held and it was determined that a data file option would require the least time but resources for the project were scarce. Also the IAS could not provide the detail information in a data file until January, 1992, which came after the implementation of the PC system. As a result, the CPCS data file was not pursued. The standards in Table 3-8 were put in place for the manual cash letter entries.

These changes permitted cash letter items to be matched automatically using the FRB code and transaction amount as the matching criteria. Some of the factors that complicated the manual process continued to complicate the automated process. However implementation of the standards made the manual portion of the process much faster and more accurate, and provided an audit trail that was absent prior to this project. Exception items are reported to EVOC and WVOC in two groups, original and return letter, which is facilitated by the use of report groups WCL, ECL, WRET, and ERET.
### Table 3-8

GL Requirements for Cash Letters

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Description 1 Field (immediate cash letters at EVOC)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>1515 - Incoming letters from FRB</td>
</tr>
<tr>
<td></td>
<td>1540 - Savings Bonds redemptions</td>
</tr>
<tr>
<td></td>
<td>1524 - Government Checks</td>
</tr>
<tr>
<td></td>
<td>1523 - Postal Money Orders</td>
</tr>
<tr>
<td></td>
<td>o Center &quot;EV&quot;</td>
</tr>
<tr>
<td></td>
<td>o Cash letter description - dependent on FRB code:</td>
</tr>
<tr>
<td>1528</td>
<td>Description 1 Field (department outgoing and incoming)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Center &quot;EV&quot; or &quot;WV&quot;</td>
</tr>
<tr>
<td></td>
<td>o Department Name(&quot;RES&quot;)</td>
</tr>
<tr>
<td></td>
<td>o 4-digit FRB reference number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;C/L ADJ&quot;</td>
</tr>
<tr>
<td></td>
<td>o FRB cash letter date</td>
</tr>
<tr>
<td>1528</td>
<td>Description 1 Field (adjustment to incoming return letter)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Center &quot;EV&quot; or &quot;WV&quot;</td>
</tr>
<tr>
<td></td>
<td>o Department Name (&quot;EXC&quot;)</td>
</tr>
<tr>
<td></td>
<td>o 4-digit FRB reference number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;CLR P (D or C)&quot;</td>
</tr>
<tr>
<td></td>
<td>o Date on adjustment form</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Short ABA Number</td>
</tr>
<tr>
<td></td>
<td>- DB Richmond  0141</td>
</tr>
<tr>
<td></td>
<td>- DB Hampton Roads  1032</td>
</tr>
<tr>
<td></td>
<td>- DB Northern Virginia  0760</td>
</tr>
<tr>
<td></td>
<td>- DB Shenandoah Valley  0156</td>
</tr>
<tr>
<td></td>
<td>- DBNA  0054</td>
</tr>
<tr>
<td></td>
<td>- DB Washington  0122</td>
</tr>
<tr>
<td></td>
<td>- DB Maryland  0320</td>
</tr>
<tr>
<td>1528</td>
<td>Description 1 Field (adjustment to outgoing return letter)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Center &quot;EV&quot; or &quot;WV&quot;</td>
</tr>
<tr>
<td></td>
<td>o Department Name (&quot;EXC&quot;)</td>
</tr>
<tr>
<td></td>
<td>o 4-digit FRB reference number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;ADJ TO FED&quot;</td>
</tr>
<tr>
<td></td>
<td>o Date on adjustment form</td>
</tr>
<tr>
<td>3005</td>
<td>Description 1 Field (incoming return letter)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Center &quot;EV&quot; or &quot;WV&quot;</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;EXC&quot;</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Short ABA Number</td>
</tr>
<tr>
<td>FRB Code</td>
<td>Required Entry</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>5704</td>
<td>Description 1 Field (for manual reclamations)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Department Name (&quot;EXC&quot;)</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;RECLAMATION&quot;</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o 10-digit Customer Account Number</td>
</tr>
<tr>
<td>3005</td>
<td>Description 1 Field (for FRB returns)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Center &quot;EV&quot; or &quot;WV&quot;</td>
</tr>
<tr>
<td></td>
<td>o Department Name - &quot;EXC&quot;</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;FED RETURNS&quot;</td>
</tr>
</tbody>
</table>
FS – Food Stamps

Federal food stamps are redeemed by the FRB. Dominion Bank’s WVOC is the largest processor of food stamps in the state of Virginia. Food stamps are collected at the branch offices and sent to either WVOC or EVOC where they are batched and returned to the FRB as a cash letter for collection on a daily basis. Each operations center makes one entry to the GL each day for the total amount of food stamps collected. Therefore two entries for food stamps appear on the GL each day. The GL description for food stamps transactions was usually not discernable from the other transactions.

On the FRB statement, food stamps appear under FRB transaction code 0804 – Food Stamps. There are two entries, one from each operations center, for the total amount of food stamps received from each operations center. After posting the cash letter to our account, the cash letter amount is verified. Frequently the FRB finds that the cash letter amount is erroneous and must make an adjustment to the account. These adjustments appear on the statement and in an Advice of Adjustments report that is sent daily with other supplementary reports. The Advice of Adjustment report is sent to the Research & Exceptions departments at EVOC and WVOC who make the appropriate adjustment entry to the GL once the adjustment amount has been verified. Adjustments are reported under code 0808 – Treasury Adjustment. Table 3-9 shows the standards established for reporting food coupon transactions on the GL.
Table 3-9
GL requirements for Food Stamps

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0804</td>
<td>Description 1 Field (for normal food stamp transactions)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;FOOD COUPONS&quot;</td>
</tr>
<tr>
<td></td>
<td>o Center - &quot;EV&quot; or &quot;WV&quot;</td>
</tr>
<tr>
<td>0808</td>
<td>Description 1 Field (Food Coupon adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Center - &quot;EVOC&quot; or &quot;WVOC&quot;</td>
</tr>
<tr>
<td></td>
<td>o Department Name - &quot;RES&quot;</td>
</tr>
<tr>
<td></td>
<td>o 4-digit FRB reference number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;FC ADJ&quot;</td>
</tr>
<tr>
<td></td>
<td>o FRB Food Coupon adjustment date</td>
</tr>
</tbody>
</table>
The FRB code and transaction amount are used as the matching criteria. Exception items are reported to the operations centers using report groups EVOC and WVOC.

**CPN - Bond Coupons**

Bond coupons payments refer to the payment of interest on U.S. Treasury bonds held by customers. A bond holder redeems a coupon by clipping it from the bond and taking it to any branch office for payment. If the bond was purchased through Dominion Bank then the coupons are processed by Dominion Trust Company (DTC). If not, then the coupons are collected at the branch offices and sent to the appropriate operations centers where they are batched and sent to the FRB for processing. The GL entries for bond coupons are made in total for the day by DTC and each operations center. A unique collection number is assigned to each coupon cash letter. The GL entry typically contained a description such as "OUTGOING COUPONS" in the Description 1 field.

The FRB reports bond coupon activity under FRB code 4001 - Coupon Collection. The amounts reported are the daily totals received by the FRB from DTC and each operations center. In addition the FRB reports a 4-digit collection number on the statement in the Reference field which is a number assigned by Dominion Bank.

The matching process was complicated by two factors. The bond coupons processed by DTC were often reported to the GL one day after they were reported on the FRB statement. This delay was due to the processing procedures in place at DTC and was eventually eliminated. The second factor that complicated the matching process was the premature reporting of coupon payments by the FRB. Often the FRB will report bond coupons that are not due until a future date. When DTC receives these premature coupons they are returned to the FRB and the account
is adjusted within a few days. When the actual coupon date arrives, the coupon payments are again reported on the FRB statement. The return of coupons and their appearance on the FRB statement several times complicates the matching process and is due to oversight on the part of the FRB. Table 3-10 lists the standards put in place for reporting bond coupon transactions on the GL.

These modifications to the GL entries permitted automatic matching of bond coupon transactions using the FRB code, collection number, and transaction amount as the matching criteria. The use of report groups WVCPN and TRCPN facilitates the reporting of exception items to the operations centers and DTC respectively.
### Table 3-10

GL requirements for Bond Coupons

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>Description 1 Field (for WVOC coupon letters of settlement)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Date of Coupon Letter</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (WVOC)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Collection Number</td>
</tr>
<tr>
<td>4001</td>
<td>Description 1 Field (for TRUST bond coupon adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Date of Coupon Letter</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (TRUST)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Collection Number</td>
</tr>
<tr>
<td>4001</td>
<td>Description 1 Field (for bond coupon adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Date of Coupon Letter</td>
</tr>
<tr>
<td></td>
<td>o Amount of original coupon letter</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (TRUST or WVOC)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Collection Number</td>
</tr>
</tbody>
</table>
FTS - Funds Transfer System

The transfer of funds between Dominion Bank and other FRB-member banks is one of the many services provided by the FRB. By wiring funds into and out of the member banks' reserve accounts, the transfer of funds between accounts at the FRB is an easy and efficient means of transferring funds between banks. The system used by Dominion for the wire transfer of funds is the Funds Transfer System (FTS) which is directly tied to the FRB's accounting system. Approximately 500 transactions pass through the system daily. To eliminate the need for each of these transactions to be entered on the GL, an automated interface was developed that totals all debits and credits and passes these total amounts to the GL each night. Therefore one debit and one credit are passed to the GL each day from FTS. These entries are easily identified by their description on the transaction journal.

Due to improper implementation and usage of FTS, the daily debit and credit totals included transactions that did not belong in the FRB account and complicated the manual reconcilement process. Three categories of transactions comprise the numbers interfaced to the GL. The first category of transactions are "Wires In" and "Wires Out." "Wires In" are transfers of funds coming into the FRB from other banks; "Wires out" are transfers of funds from our FRB account to another bank. These transactions are reported under code 1001 - Transfer of Funds on the FRB statement as a total debit and total credit.

The second set of FTS transactions that were included in the interfaced amounts are internal transfers of funds. Internal transfers are used by various areas of the bank to make a debit or credit to GL account 1060 for the transfer of funds within the bank. These transactions are internal to Dominion Bank and are not reported by the FRB. Internal
transfers must be validated against other GL entries and are a match of GL to GL and must be reconciled manually. These transactions did not involve the FRB and should never have passed through account 1060. Once these problems were revealed the accounting department banned the use of FTS for internal transfers involving account 1060. This eliminated one problem area for FTS.

The third category of FTS transactions are "In Wires" and "Out Wires." These are transfers of funds with other banks where the FRB is not used as an intermediary in the transaction. We discovered that the "In Wires/Out Wires" were being used by the Funds Management area to exchange funds between banks for MBS trades that did not clear through the FRB. These transactions could be easily matched against offsetting GL entries with the proper reporting from the MBS system. Therefore these transaction required a GL-to-GL match that would have to be performed manually.

The manual matching process for FTS required a comparison of the FTS interface report to the FRB statement. The total debit and total credit amounts for "Wires In/Wires Out" could be matched against the debit and credit totals for FRB code 10019. Since internal transfers and "In Wires/Out Wires" did not involve the FRB, these totals were not reported on the FRB statement. Therefore the amounts interfaced to GL from FTS were wrong because they included activity not reported by the FRB.

Several FTS reports were used in the matching process. Reports 520 and 521 provided a listing, broken out by debits and credits, of each wire transfer transaction that occurred the prior day. Report 565, an interface report, showed the amounts posted to each GL account each day, with total debits and credits for each of the three transaction types described above. Using Report 565 it was possible to eliminate any "In
Wire/Out Wire" activity from the interfaced amounts, leaving only "Wire In/Wire Out" activity to match against the FRB statement. On days when "In Wire/Out Wire" transactions are posted to account 1060, the GL totals do not match the totals reported on the FRB statement and Report 565 must be used to verify the "Wire In/"Wire Out" totals.

There is one more factor with regard to FTS that is worthy of discussion. The manual matching process was based on the comparison of total debit and total credit amounts, which are made up of approximately 500 transactions. If any one of these transactions was reported incorrectly by either side (FRB or GL) then the totals would not match. When this occurred, each individual transaction had to be matched to determine which item caused the outage. I was informed that this occurred only rarely, but when it did occur, it caused significant delays in the balancing process. I felt that the issue should be pursued. The FRB was providing the accounting department with a daily report listing the individual wire transfer transactions that made up the totals reported on the statement. I investigated the possibility of replacing the total amounts with the transaction detail in the data file. I learned that the IAS system could provide this detail and that the programmer in Richmond could easily make the change with a few day's notice.

Next I looked at making a similar change on the FTS system. I studied the reports generated by FTS and determined that if a file could be produced containing the needed data then FTS matching could occur on a transaction-level basis.

After studying FTS reports 520, 521 and the detailed data file being provided by IAS, I determined that a transaction was uniquely identified by the combination of debit/credit code, transaction amount, and Sending Bank number (which is the 9-character ABA number of the other bank involved in the
transaction). The Sending Bank number was also being reported by the FRB in the Offset Transit Routing number field (BANK_DFI in the data base file). Therefore individual wire transfer transactions could be matched on a daily basis to eliminate the possibility of having to do it manually when totals did not match. Table 3-11 contains the file layout of the FTS file requested on the basis of this analysis.

This file was incorporated into the original version of the automated matching program. It was later eliminated because the size of the file caused processing delays. In a trade-off between these delays and the necessity of matching on an individual transaction, faster processing time was chosen and this segment of the program was deleted. The original method of matching based on total debit and credit activity was reinstated. The detailed matching at the transaction level will be used in the mainframe system.
Table 3-11

FTS File Layout

<table>
<thead>
<tr>
<th>Character Position</th>
<th>Field Name</th>
<th>Type &amp; Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>TRANS_CODE</td>
<td>A4</td>
<td>FRB Transaction Code. Wires In/Wires Out are identified by &quot;1001,&quot; Internal Transfers and In Wires/Out Wires are identified with &quot;0000&quot; a dummy transaction code since they are not reported by the FRB.</td>
</tr>
<tr>
<td>5-11</td>
<td>SEQ_NO</td>
<td>A7</td>
<td>For debits, the FRB Sequence number is reported, for credits the FTS Sequence number is reported.</td>
</tr>
<tr>
<td>12-13</td>
<td>TYPE</td>
<td>A2</td>
<td>Short alpha code identifying the type of transaction. &quot;IW&quot; for interface Wires In, &quot;OW&quot; for interface Wires Out, &quot;IT&quot; for Internal Transfers.</td>
</tr>
<tr>
<td>14-15</td>
<td>DEBCRE</td>
<td>A2</td>
<td>Debit/credit indicator, &quot;10&quot; for debit, &quot;60&quot; for credit.</td>
</tr>
<tr>
<td>16-24</td>
<td>ROUTE</td>
<td>A9</td>
<td>ABA number of other bank/broker involved in transaction. This provides a match with FRB BANK_DFI field.</td>
</tr>
<tr>
<td>25-32</td>
<td>DATE</td>
<td>D8</td>
<td>Effective date of entry</td>
</tr>
<tr>
<td>33-48</td>
<td>AMOUNT</td>
<td>99(13).99</td>
<td>Amount of transaction</td>
</tr>
<tr>
<td>49-100</td>
<td>FILLER</td>
<td>A51</td>
<td>Blank spaces to pad record to 100 characters.</td>
</tr>
</tbody>
</table>
Investments

The investments group includes all transactions involving U.S. Treasury securities. The Treasury securities involved are bills, notes bonds, and mortgage-backed securities. These security transactions occur in several areas of the bank which are the basis for the report groupings AIM, MBS, MBSPD, MMII and TRUST. The report group OTH is used to identify investment transactions whose origin can not be determined. These categories of transactions are discussed separately in the following sections.

AIM - Automated Investment Management System

The Funds Management area of the bank is responsible for managing the bank's portfolio and for providing trading and accounting services for correspondent and affiliate bank portfolios. MBS's are processed on an in-house developed PC system. All other types of securities are processed on AIM, which is an in-house developed mainframe system. The AIM system does not automatically interface to the GL system, therefore GL entries are made manually by the accounting staff using reports generated by the AIM system. During this project, the AIM system was in the process of being converted to MMII which has an automated interface to the GL system. Therefore an automated interface from AIM to GL was not pursued.

AIM transactions involving U.S. Treasury securities pass through the FRB account and appear on the FRB statement. These transactions include the purchase and sale of U.S. Treasury and agency securities, municipal bonds or FRB stock, as well as the maturities, calls, and interest payments on these securities. In addition, a special type of security, Treasury Tax and Loan (TT&L) could be traded. On the FRB statement these transactions are reported under the following
FRB codes by security:

0802 - TT&L Transaction
0811 - Treasury Calls
0812 - Remittance Payments
0822 - TT&L Adjustments
0825 - TT&L Reversals
2001 - Purchase/sale of Treasury securities
2701 - Maturities of Treasury securities (principal amount)
2702 - Maturities of Treasury securities (interest amount)
4001 - Purchase/sale/interest on municipal bonds
4002 - Municipal bond collection (principal amount)
6001 - Interest on FRB stock
6002 - Purchase/sale of FRB stock
6003 - Adjustment to FRB stock

An Advice of Credit report containing transaction detail for the security transactions in codes 2001, 2011, 2701 and 2702 was provided daily by the FRB and could be used to identify specific securities by their CUSIP number. On the FRB statement and Advice of Credit, the AIM transactions were interspersed with other investment transactions from MMII, Trust and the MBS system.

To perform the manual matching process, reports from the AIM system were sent to the accounting department where GL entries were made in summary by day. This meant that not only was there a delay between the time the FRB reported the transactions and when the GL entries were made, but the entries were not being made in a form compatible with the FRB statement. To perform the manual reconciliation, one had to refer back to the AIM reports to determine which securities and amounts comprised the GL entry. This was a very time consuming process. The process was further confused by the fact that the AIM transactions were not differentiated from the MMII, Trust and MBS transactions on the FRB statement. To be consistent with the FRB reporting standards for security
transactions, it was necessary to report the daily transaction amount for each security. Table 3-12 shows the standards put in place for AIM transactions. These standards for manual entries represented an interim measure because the conversion to MMII would eliminate the need for manual entries. The conversion was completed in March, 1992.
Table 3-12
GL requirements for AIM

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0802</td>
<td>Description 1 Field (TT&amp;L transactions)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>0802 - TT&amp;L Transaction</td>
</tr>
<tr>
<td></td>
<td>0811 - Treasury Call</td>
</tr>
<tr>
<td></td>
<td>0812 - Remittance Payment</td>
</tr>
<tr>
<td></td>
<td>0813 - TT&amp;L Direct Investment</td>
</tr>
<tr>
<td></td>
<td>0822 - TT&amp;L Adjustment</td>
</tr>
<tr>
<td></td>
<td>0825 - TT&amp;L Reversal</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Transaction date</td>
</tr>
<tr>
<td>2001</td>
<td>Description 1 Field (Purchase/sale of Treasuries)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (AIM)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
<tr>
<td>2701</td>
<td>Description 1 Field (Treasury maturities - principal)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (AIM)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
<tr>
<td>2702</td>
<td>Description 1 Field (Treasury maturities - interest)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (AIM)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
<tr>
<td></td>
<td>Description 1 Field (Municipals)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>0401 - Purchase/sale</td>
</tr>
<tr>
<td></td>
<td>0402 - Collection</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (AIM)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
</tbody>
</table>
Table 3-12 (Cont)

GL requirements for AIM

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Description 1 Field (FRB Stock)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>6001 - Interest</td>
</tr>
<tr>
<td></td>
<td>6001 - Purchase/sale</td>
</tr>
<tr>
<td></td>
<td>6003 - Adjustment</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (AIM)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
</tbody>
</table>
MBS and MBSPD - Mortgage-Backed Securities System

Portfolio transactions for Dominion Bank and its correspondent and affiliate banks involving MBS's are processed on the MBS system in the Funds Management area. MBS's are offered by various federal government mortgage agencies such as the Federal National Mortgage Association (FNMA), Federal Home Loan Mortgage (FHLM), Government National Mortgage Association (GNMA), and others and backed by home mortgage loans. An MBS is a group of mortgage loans that are sold by the various agencies to investors. The investors receive the proceeds of the monthly mortgage payments (principal and interest). The mortgage payments are made by the mortgage holder to the lending agency on the designated payment date each month and the agency in turn passes this money on to the investor. The process of paying down the principal and interest on a mortgage on a monthly basis is called a "paydown" and is designated by the report group MBSPD. The report group used to specify MBS trading activity is MBS.

Since MBS's are government securities, they are bought and sold through the FRB. Therefore the FRB is involved in the trading of MBS's and in the monthly paydown process. Whenever Dominion conducts an MBS trade with another bank or broker, the securities are cleared through our account at the FRB. Therefore this trade and paydown activity appears on our FRB statement. A study of the FRB statement revealed that MBS activity is reported under three transaction codes. Trade activity is reported under code 2011, reversals of trades in 2019, and paydowns in 2702. MBS securities are traded in large groups generally in many millions of dollars. Therefore the MBS system processes trades on a trade basis which includes many securities. The system produces reports showing the amount of each security traded and then provides a trade
total. These reports were sent to the accounting department where the GL entries are made by trade.

For paydown activity, a similar method is used. The various security types, such as FNMA and GNMA pay down on a certain date each month. Therefore a single GL entry is made on each pay day for the total of all securities that paid down that day. This GL entry is made by the Funds Management area. A report detailing the amount paid down for each security on the MBS system is sent to the accounting department to facilitate the reconciliation process.

The manual reconciliation process was difficult and time consuming because the GL entries were made on a trade or security type basis while the FRB statement reported transaction amounts by security. The manual process was further complicated by an issue related to the transfer of funds between banks involved in MBS trades. "Pair-off" and "Round Robin" trades sometimes caused problems in the reconciliation process because certain securities did not clear though the FRB as expected. When this occurs the matching procedure becomes much more complex. GL entries are made for the total amount of the trade, but only a subset of these securities clear through the FRB and are reported on the FRB statement. The securities that clear through the FRB can be matched against the Funds Management detailed reports in the usual manner. For the securities that do not clear through the FRB, the Funds Management area determines which banks/brokers owe us money and the amount, and to whom we owe money and the amount as a result of the security swaps that took place. They then use the FTS system to wire money to other banks if needed, or inform banks how to wire in money that is owed to us. Therefore a wire of funds to or from a bank may involve money from several unrelated trades that occurred on the same day. Tracing the transactions that
occurred on FTS to the trade that they are a part of is a time consuming process and was an obstacle to the automation project but could not be eliminated due to the nature of trading activity.

The goals for the automation of MBS trades and paydowns was to report the information on the GL system on a security basis just as the information was being reported by the FRB. However, it was unrealistic to expect the Funds Management area and the accounting department to replace their trade totals with hundreds of entries at the security level. I pursued the idea of obtaining a data file from the MBS system with the security-level detail needed to perform matching at the security level. I was testing the data file when the programmer took a 4-month maternity leave. During her absence it was decided for the long-term, an automated interface to the GL containing transaction data at the security level should be developed. The plan was to have this interface in place in conjunction with the implementation of the mainframe reconciliation system. Thus the pursuit of a data file was abandoned in favor of the long-term goal of an automated interface. As a result, we were left with maintaining the trade level manual GL entries with the using the standards shown in Table 3-13. These modified entries did not permit automated matching of the MBS trades and paydowns, but made the recognition of MBS activity much easier. The manual reconciliation continued with the use of the detailed MBS reports.
Table 3-13
GL requirements for MBS and MBSPD

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>Description 1 Field (for purchase/sale of MBS's)</td>
</tr>
<tr>
<td></td>
<td>- FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>- Transaction description</td>
</tr>
<tr>
<td></td>
<td>- Department Name (MBS)</td>
</tr>
<tr>
<td>2019</td>
<td>Description 1 Field (for reversal of purchase/sale)</td>
</tr>
<tr>
<td></td>
<td>- FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>- Transaction description</td>
</tr>
<tr>
<td></td>
<td>- Department Name (MBS)</td>
</tr>
<tr>
<td>2702</td>
<td>Description 1 Field (paydowns)</td>
</tr>
<tr>
<td></td>
<td>- FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>- Transaction description</td>
</tr>
<tr>
<td></td>
<td>- Department Name (MBSPD)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>- Security type (GNMA, FNMA, FHLM)</td>
</tr>
</tbody>
</table>
MMII - Money Market II

The MMII system is used to process transactions and to perform the accounting and safekeeping functions for a wide variety of investment products. The transactions include the purchase and sale of securities for the customers of Dominion Investment Banking, Inc. (which is the broker/dealer subsidiary of Dominion Bank) and beginning in March, 1992, the portfolio accounts of Dominion Bank and its affiliate and correspondent banks. MMII is a mainframe system housed at the vendor’s facility in Waltham, Massachusetts. Dominion is a time-sharing customer of the system which means that we pay a monthly fee to use the system but the hardware and software is maintained at the vendor’s site.

U.S. Treasury bills, notes and bonds are issued by the FRB and the purchase, sale, maturity, and interest payments on these securities passes through the FRB account. The MMII system sends interfaced entries to all affected GL accounts each night following the update cycle. The entry posted to each account is the net amount of all debits and credits for the prior day. In addition to this interfaced entry, the MMII operations staff makes manual entries for adjustments and some MBS paydowns which are not interfaced to the GL because they are back-dated more than one month.

The detail supporting the interfaced entry for all GL accounts is provided by a MMII report which is sent daily to the accounting department. In the report, transaction amounts are reported separately for each customer holding. On the FRB statement, the transactions are reported by security. These transactions appear under the following FRB codes:

2001 - Purchase/sale of Treasury securities
2011 - Purchase/sale of Non-Treasury securities
2019 - Purchase/sale of Treasury securities - Reversal
2020 - Purchase/sale of Non-Treasury securities -
Reversal
2701 - Maturities of Treasury securities (Principal amount)
2702 - Maturities of Treasury securities (Interest amount)

These codes correspond to certain transaction types and security types on the MMII system. Therefore the matching process requires that the transactions that are reported by customer holding on the MMII detail report be subtotalled by security and by transaction type before they are matched against the FRB statement.

While the GL entry for MMII transactions is easily identified by its Source Code and Description 1 field on the transaction journal, the manual reconciliation process was dependent upon the transaction level detail contained in the MMII report. Unfortunately the sort order of the report did not facilitate the summation of transactions by transaction type and security. The transaction detail from the MMII system was needed to automate this portion of the account. Modifying the GL interface so that it would send the detail instead of summarized entries was considered infeasible because the large number of entries would slow the daily processing of the GL system. Thus I considered the possibility of obtaining the needed detail in a data file.

After several weeks of study I was able to establish the connection between FRB codes and the MMII transaction type and security types. This relationship is summarized in Table 3-14. Having defined the relationship between FRB codes and MMII transactions I asked the vendor for a cost estimate to develop a data file in the format shown in Table 3-15. The vendor responded to this request with a cost estimate of approximately $5000.

The accounting managers were reluctant to pay for
development of the report data file because the mainframe reconciliation system might require a different file layout. Thus we were left with the current manual process of comparing the MMII report to the FRB statement. For adjustments and MBS paydowns Table 3-16 shows the standards put in place for the manual GL entries.
### Table 3-14

**Relationship between FRB Codes & NMII Identifiers**

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Transaction Type</th>
<th>Security Type</th>
<th>Principal, Int., or Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>REPO</td>
<td>RSA</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>RSAL</td>
<td>RPA</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>MAT</td>
<td>RSA, RPA</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>BUYP</td>
<td>BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>SELF</td>
<td>BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>CSAL</td>
<td>BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>SSAL</td>
<td>BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td>2019</td>
<td>Reversals of the above</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>BUYP</td>
<td>all but BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>SELF</td>
<td>all but BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>CSAL</td>
<td>all but BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>SSAL</td>
<td>all but BILL, NOTE, BOND, ZERO</td>
<td>Both</td>
</tr>
<tr>
<td>2020</td>
<td>Reversals of the above</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2701</td>
<td>SMAT</td>
<td>all but RPA, RSA</td>
<td>Principal</td>
</tr>
<tr>
<td></td>
<td>MAT</td>
<td>all but RPA, RSA</td>
<td>Principal</td>
</tr>
<tr>
<td></td>
<td>AABF</td>
<td>all</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>AASF</td>
<td>all</td>
<td>Both</td>
</tr>
<tr>
<td>2702</td>
<td>APRN</td>
<td>all</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>PRNS</td>
<td>all</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>INT</td>
<td>all</td>
<td>Interest</td>
</tr>
<tr>
<td></td>
<td>SINT</td>
<td>all</td>
<td>Interest</td>
</tr>
<tr>
<td></td>
<td>MAT</td>
<td>all but RPA, RSA</td>
<td>Interest</td>
</tr>
<tr>
<td></td>
<td>SMAT</td>
<td>all but RPA, RSA</td>
<td>Interest</td>
</tr>
</tbody>
</table>
### Table 3-15

**MMII File Layout**

<table>
<thead>
<tr>
<th>Character Position</th>
<th>Field Name</th>
<th>Type &amp; Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>DEB_CRE</td>
<td>A2</td>
<td>Debit/credit code, &quot;60&quot; for credit, &quot;10&quot; for debit</td>
</tr>
<tr>
<td>3-8</td>
<td>DATE</td>
<td>A6</td>
<td>Effective date, &quot;MMDDYY&quot;</td>
</tr>
<tr>
<td>9-23</td>
<td>AMOUNT</td>
<td>9(13)v99</td>
<td>Amount, no commas, no &quot;$&quot;</td>
</tr>
<tr>
<td>24-28</td>
<td>FRB_CODE</td>
<td>A5</td>
<td>4-character FRB code</td>
</tr>
<tr>
<td>29-32</td>
<td>MMII</td>
<td>A4</td>
<td>&quot;MMII&quot;</td>
</tr>
<tr>
<td>33-41</td>
<td>CUSIP</td>
<td>A9</td>
<td>9-character CUSIP number</td>
</tr>
<tr>
<td>42-45</td>
<td>TTYPE</td>
<td>A4</td>
<td>MMII transaction type code</td>
</tr>
<tr>
<td>46-49</td>
<td>STYPE</td>
<td>A4</td>
<td>MMII security type code</td>
</tr>
</tbody>
</table>

### Table 3-16

**GL Requirements for MMII**

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>2702</td>
<td>Description 1 Field (paydowns)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (MMII)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
<tr>
<td>2019</td>
<td>Description 1 Field (adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (MMII)</td>
</tr>
</tbody>
</table>
Trust - Dominion Trust Company

Dominion Trust Company is the Dominion Bank subsidiary that provides trust services to customers. Trust services include the management and disbursement of money and securities set up as trust funds. The management of these trust funds includes the purchase and sale of various types of securities including U.S. Treasury securities. Therefore DTC is another area of the bank where FRB transactions originate.

The Mellon Bank trust system is used by DTC to perform the transaction processing and accounting for customer trust funds. For account reconciliation, the system generates daily reports showing the total transaction amount by security for purchase, sale, maturity, paydown and interest payment transactions. These amounts were totalled for the day’s activity and entered int the GL as single debit and credit each day.

On the FRB statement, Trust transactions appeared under codes 2001, 2701, and 2702. Using these codes the FRB reported DTC transactions by security in the same manner as all other investment transactions.

DTC transactions presented another case where a summarized GL entry was being matched against FRB security-level transactions. Two options were considered for automating the matching process for these transactions. One alternative was to have the Mellon Bank system provide the detail data needed for the PC program by either developing a detailed interface from Mellon system to the GL, or to provide the data in a data file that could be read int the PC system. The second alternative was to modify the manner in which manual GL entries were being made. After several meetings with the Vice President of DTC it was determined that the manual entries could be made at the security level to conform with the FRB reporting standards. This decision was based on the fact that
DTC had completed a conversion from the Mellon Bank system to the SEI system and then decided to convert back to the Mellon Bank system after a year of struggles with the SEI system. This conversion and reconversion had taken its toll on the entire DTC staff and since the reconversion was still in progress no additional systems issues could be addressed at that time. Table 3-17 shows the GL requirements for the DTC entries to the GL. With these standards in place, it was possible to match DTC transactions using FRB code and transaction amount as the matching criteria.
<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>Description 1 Field (for purchase/sale of Treasuries)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (TRUST)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
<tr>
<td>2701</td>
<td>Description 1 Field (maturities excluding interest)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (TRUST)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
<tr>
<td>2702</td>
<td>Description 1 Field (paydowns (P&amp;I) &amp; interest payments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Department Name (TRUST)</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o CUSIP Number</td>
</tr>
</tbody>
</table>
SB - U.S. Savings Bonds

Dominion Bank branch offices sell U.S. Savings bonds issued by the U.S. Treasury. Each bond sale is entered into the FRB's Regional Delivery System (RDS) which is directly accessible through a computer terminal at each branch office. Since the FRB issues the savings bonds, these transactions pass through the FRB account.

At the close of each business day each branch office totals the amount of bonds sold that day and makes an entry to the GL. The FRB totals all activity by branch office and reports the total amount sold by each branch each day under FRB code 7001 - Savings Bonds Issued. In addition, a unique Issuing Agent number identifying each branch is reported in the Reference field of each transaction. Adjustments to savings bonds transactions are reported under code 7004 - Savings Bonds Adjustments.

Two problems were identified in the manual reconciliation of savings bonds. The first problem was an inconsistency in the cut-off times used by branch offices as the basis for their GL entries. The FRB used a 2 p.m. - 2 p.m. window for transactions. Any transactions occurring after 2 p.m. were reported the next day. Some Dominion branch offices used this window as the basis for their GL entries and others did not. Once this inconsistency was identified as the cause of reconciliation problems for savings bonds all branch offices were notified and asked to use the same transaction window.

The second problem encountered in reconciling savings bonds was the use of pricing factors for the redemption of bonds. Savings bonds can be redeemed at any branch office. The tellers use a pricing factor table to determine the maturity value of the bond. Careless errors occurred frequently when the incorrect factors were used for redemption. These errors affected the balancing of cash letters because redeemed
bonds were returned to the FRB along with all other cash letters. This problem was addressed at the branch office level and rules were put in place for the redemption of savings bonds.

With the two problems under control, automation of the matching process for savings bonds was not difficult. Table 3-18 shows the standards put in place for the GL entries. With these standards in place automatic matching could occur using FRB code, transaction amount, and Issuing Agent number as the matching criteria. Exceptions are reported to each region by the use of report groups SBGH, SBNA, SBNV, SBR and SBSV.
Table 3-18

GL Requirements for Savings Bonds

<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>7001</td>
<td>Description 1 Field (for sales and redemptions)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Number of savings bonds sold/redeemed</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td></td>
<td>o Branch Name</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Issuing Agent number</td>
</tr>
<tr>
<td>7004</td>
<td>Description 1 Field (for adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;SB BD ADJ&quot;</td>
</tr>
<tr>
<td></td>
<td>o Date of original entry</td>
</tr>
<tr>
<td></td>
<td>o Branch name</td>
</tr>
<tr>
<td></td>
<td>Description 2 Field</td>
</tr>
<tr>
<td></td>
<td>o Issuing Agent number</td>
</tr>
</tbody>
</table>
Other

The Other balancing group code is assigned to transactions representing fees charged by the FRB and for transactions whose origin cannot be identified because of an insufficient description.

The FRB fee transactions can be matched against the transactions reported under code 8403 - FRB Service Charges. Prior to this project the GL entries for FRB fees were being prepared by numerous areas of the bank including Research & Exceptions, Commercial Services, and Account Exceptions. To improve the controls associated with the FRB account, the accounting department assumed responsibility for making all GL entries related to FRB fees. Table 3-19 shows the standards put in place for GL entries associated with FRB fees. With these standards in place it was possible to automate the matching process for FRB fees using FRB code and transaction amount as the matching criteria.
<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>8403</td>
<td>Description 1 Field (for FRB fee transactions)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
</tbody>
</table>
CB - Correspondent Banks

Correspondent banks, for which Dominion Bank provides many services, have their FRB accounts reconciled by Dominion. These banks' FRB statements are included with Dominion's statement, but are segregated by bank. Correspondent banks are involved in five types of transactions with the FRB: ACH, cash letters, savings bonds, TT&L and FRB fees. These transactions are assigned report group codes CBACH, CBCL, and CBOTH. The GL entries for correspondent banks are made by operations areas in Roanoke, Richmond, and Greater Hampton Roads depending upon the location of the correspondent bank.

The FRB reports correspondent bank activity using the same conventions discussed for ACH, CL, and OTH. In the FRB data file used in the automated matching system the 9-character ABA field identifies the correspondent bank to whom the transaction belongs. Table 3-20 shows the GL requirements for correspondent bank entries. With these standards in place it was possible to automatically match CB transactions using FRB code and transaction amount as the matching criteria.
<table>
<thead>
<tr>
<th>FRB Code</th>
<th>Required Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1531</td>
<td>Description 1 Field (for daily CB cash letters)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;C/L&quot;</td>
</tr>
<tr>
<td>1531</td>
<td>Description 1 Field (for CB cash letter adjustments)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;C/L ADJ&quot;</td>
</tr>
<tr>
<td>1531</td>
<td>Description 1 Field (for CB cash letter returns)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;RETURNS&quot;</td>
</tr>
<tr>
<td>7001</td>
<td>Description 1 Field (for CB savings bonds transactions)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;SAV BONDS&quot;</td>
</tr>
<tr>
<td>0802</td>
<td>Description 1 Field (TT&amp;L, Lee Bank and Trust, Bedford Bank)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number (0416 or 7255)</td>
</tr>
<tr>
<td></td>
<td>o Transaction description</td>
</tr>
<tr>
<td>0815</td>
<td>Description 1 Field (TT&amp;L late charges)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number (0416 or 7255)</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;TT&amp;L LATE CHARGES&quot;</td>
</tr>
<tr>
<td>5701</td>
<td>Description 1 Field (for CB ACH)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;ACH DEBIT&quot; or &quot;ACH CREDIT&quot;</td>
</tr>
<tr>
<td>8403</td>
<td>Description 1 Field (for FRB charges)</td>
</tr>
<tr>
<td></td>
<td>o FRB transaction code</td>
</tr>
<tr>
<td></td>
<td>o 4-digit respondent bank number</td>
</tr>
<tr>
<td></td>
<td>o Transaction description - &quot;FED BILL&quot;</td>
</tr>
</tbody>
</table>
Compliance

The success of the automated matching program was dependent upon the successful implementation of the GL standards for manual entries. Overall compliance was tracked for approximately six weeks. All non-compliant entries were recorded and the originating areas were contacted weekly by telephone. At the end of the six weeks all areas of the bank were in compliance with the standards however exceptions will always occur because of keying errors.

Program Design

An automated matching system was pursued because it had the potential to eliminate a significant portion of the manual effort required in the daily reconciliation process. The short-term PC program had provided a means of keeping track of exception items and provided the ability to report the items back to their originating areas. However, the matching process was still being done manually, and the exception items had to be added and deleted from the data base files on a daily basis. This system made the balancing process vulnerable to out-of-balance conditions and the resulting delays because there were no controls in place to ensure that the account would not fall out of balance as items were added and deleted.

The theory behind the automated matching system is that it would use source data from the GL and FRB systems, pair-off all items possible, and leave behind the times that could not be matched. Control checks were incorporated throughout the program to ensure that no items were lost in the import, matching, or clearing functions and that the account would not fall out of balance unintentionally. Thus the benefits to be derived were the elimination of much of the manual matching, eliminating the input of exception items, and eliminating the
risk inherent in the freedom to add and delete items without control checks.

The Programmer's Documentation in Appendix C contains the system flowchart, module descriptions and report examples. A high-level description of the system is included here.

**System Overview**

The automated matching program balances the account by transaction group. In the first step the GL and FRB data files are imported into their respective data base files and each transaction is assigned to a group based on the FRB code. Any unidentified transactions are brought to the screen so that the user can assign a balancing group code. After all items have been assigned to a group, the current day's activity is merged with the exception items within each group to obtain a cumulative activity file for each group.

Certain fields in the transaction record can be modified to facilitate the matching process. For example, one record could be replaced with two records as long as the amount of the new records is equal to the amount of the original record. Automated matching is performed by group and is initiated by the user. For the selected group, the cumulative files for GL and FRB are compared using the matching criteria discussed earlier for each group. A report of matched items is generated for the user to review. The records are deleted from the data base files at the user's request, but only when the net activity is zero, ensuring that the account will not fall out of balance. Reports of exception items can be generated at any time for a specified balancing group or report group. The user can use these reports to manually match items that would not be automatically matched. Manually matched items can be cleared from the data base files by group at the user's request, but only if the net activity is zero.
Data Base Files

The system uses separate data base files for the GL and FRB transactions within each balancing group so that each group has a GL and FRB file. During the import function the GL and FRB source data files are read into separate daily activity data base files. Once all of the transactions in each file have been assigned to a group the records for each group are sent to the appropriate group file. During the automatic matching process, the GL and FRB file for each group are compared using a relate function that is based on the matching criteria for that group. Matched transactions remain in each data base file until the user clears the file.

Modules

The automated matching program is written in modules that are accessed through menus. The menu tree structure is shown in Figure 3-3. The user can proceed through the daily sequence of events by progressing step-by-step through the menu. The Programmer’s Documentation describes the function of each module.

Testing

Testing was an important factor in the development of the automated matching program because it would help to determine if the system operational requirements had been satisfied. Each program module was tested during its development phase. When all of the modules were completed, a test plan was developed to test the effectiveness of the system as a whole. The goal of the test plan was to assess total system performance in an operational environment using test data that was representative of actual data. The test script for the test plan is contained in the Programmer’s Documentation in Appendix C.
The test plan divided the program into seven cycles, representing system functions that manipulate data. The cycles are as follows:

Cycle 1 - Importation of Files/Categorization of Transactions
Cycle 2 - Assigning Group Codes
Cycle 3 - Transaction Modification
Cycle 4 - Matching Items
Cycle 5 - Clearing Transactions
Cycle 6 - Reports
Cycle 7 - Restoring to Previous Day

Test conditions were developed representing all possible scenarios that might be encountered in the operating environment. Test files were also developed containing data representative of actual data and also data representing unusual conditions.

Test Results
The test was repeated as needed for each cycle as problems were uncovered and corrected. Once all of the cycle tests were successful, a second test of the entire system was conducted to ensure overall success of the system and to provide documentation following the flow of data items from beginning to end of the process. Following this test, the accounting managers accepted the system having verified the data manipulation and soundness of system operation.

Implementation
The PC system was implemented in late December, 1991. It was necessary to convey the exception items stored on the short-term PC system to the new system. This required additional programming because of the difference in data base structure between the two systems. On the short-term PC
system, exception items were kept in four files based on transaction origination ("We Debit," "We Credit," "FRB Debit," "FRB Credit"). On the new system, there was a set of files (GL and FRB) for each balancing group. The conversion program took the data from the short-term files, sent them to GL and FRB data files using the Smart II report generator and then read the files into the new system just as it would do for actual data. Once the existing exception items had been processed by the new system, the two systems were run in parallel for one week so that the staff could be trained on the new system and to give them an opportunity to become familiar with the new reports.

The daily procedure begins with downloading the GL and FRB data files from the mainframe computer to the the PC using Arbiter communications software. These files are automatically created each night on the mainframe computer. The program is then executed, beginning with the import of the data files and the assignment of any unidentified transactions. From there, the user can modify items to facilitate the matching process. This is accomplished by referring to the detail reports that are still sent to the Accounting Department. Items can be moved between groups as needed, or can be broken down or consolidated, as long as the changes do not put the account out of balance. Following this procedure the files are matched automatically by group. Reports listing the automatically matched items can be used to verify the matches. Changes can be made during the clearing session for each group which also permits the clearing of items that were matched manually. The matching and clearing functions can be repeated as needed throughout the day. Stale item reports are sent to the originating areas on a schedule established by management. Any other reports can be generated at any point during the day.
Main Menu
A) Import files
B) Modify Group Files
C) Automated Matching by Group
D) Clearing by Group
E) Reports Menu
F) Restore Prior Day's Files
G) Back Up Files to Diskette
X) Exit

B) Modify Group Files   C) Automated Matching by Group
  A) Correspondent Banks   B) Automated Clearing House
  C) Coin & Currency   C) Coin & Currency
  D) Cash Letters   D) Cash Letters
  E) Coupon Payments   E) Coupon Payments
  F) Food Stamps   F) Food Stamps
  G) Funds Transfer System   G) Funds Transfer System
  H) Investments   H) Investments
  I) Savings Bonds   I) Savings Bonds
  J) OTHER   J) OTHER

X) Return to Main Menu

D) Clearing by Group
  A) Correspondent Banks
  B) Automated Clearing House
  C) Coin & Currency
  D) Cash Letters
  E) Coupon Payments
  F) Food Stamps
  G) Funds Transfer System
  H) Investments
  I) Savings Bonds
  J) OTHER

X) Return to Main Menu

E) Reports Menu
  A) G/L Balancing Group file for date range
  B) G/L Detailed Group file for date range
  C) FRB Balancing Group file for date range
  D) FRB Detailed Group file for date range
  E) G/L File Replacements
  F) FRB file Replacements
  G) All Outstanding Items
  H) Outstanding Items by Detailed Group
  I) Stale Items for Detailed Group
  J) Stale Items for All Groups
  K) Cover Sheet

X) Return to Main Menu

Figure 3-3
Automated Matching System Menu Tree
CHAPTER 4
RESULTS

The two-phase approach to the problem of reconciling Dominion Bank's FRB account permitted us to satisfy the critical need for improving the manual process, while allowing us to plan a long-term solution. The implementation of the short-term program provided an automated means of keeping track of exception items by allowing the addition and deletion of items in four database files representing "We Debits," "We Credits," "FRB Debits," and "FRB Credits." This eliminated the need to manually transcribe the revised list, initially comprising 1300 items, on a biweekly basis. The migration from a biweekly to a daily reconciliation would have been impossible without the implementation of this program. The short-term program also provided the ability to print reports of exception items in a variety of formats which made it possible to focus on stale items and to report only the relevant items to the originating areas which was made possible by the use of the balancing and report group codes.

The balancing phase of the process was also enhanced by the short-term program because it provided the ability to instantaneously calculate net totals of exception items. This feature helped to identify out-of-balance conditions prior to the close of business each day. The use of the balancing group codes made the balancing process more efficient because out-of-balance conditions could be identified by group, permitting faster resolution of problem conditions. Another feature of the short-term program that enhanced the reconciliation process was the ability to attach descriptive information to the exception items in the database files using a 30-character description field. Special reports could be generated as needed based on the information in this field,
which aided in the clearing of exception items.

Implementation of the automated matching program not only eliminated much of the manual effort required to reconcile the account, but provided the incentive needed to standardize GL entries and to establish procedures for reporting and resolving exception items. The automated matching program provided capabilities above and beyond those of the short-term PCsystem. The automatic matching function of the system eliminated the need to manually match every item every day. The system provided the ability to match items that appeared on different days. The approximate success rate of automatic matching for each group is shown below:

<table>
<thead>
<tr>
<th>Group</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correspondent Banks - CB</td>
<td>50%</td>
</tr>
<tr>
<td>Automated Clearing House - ACH</td>
<td>75%</td>
</tr>
<tr>
<td>Coin &amp; Currency - CC</td>
<td>80%</td>
</tr>
<tr>
<td>Cash Letters - CL</td>
<td>40%</td>
</tr>
<tr>
<td>Bond Coupons - CPN</td>
<td>75%</td>
</tr>
<tr>
<td>Food Stamps - FS</td>
<td>75%</td>
</tr>
<tr>
<td>Funds Transfer System - FTS</td>
<td>80%</td>
</tr>
<tr>
<td>Investments - INV</td>
<td>20%</td>
</tr>
<tr>
<td>Other - OTH</td>
<td>20%</td>
</tr>
<tr>
<td>Savings Bonds - SB</td>
<td>80%</td>
</tr>
</tbody>
</table>

Overall the average success rate of matching is approximately 60 percent. This means that less than half of the transactions require individual attention daily.

The automated matching system not only eliminated a significant portion of the manual matching process, but also lent credibility to the balancing process. The control checks incorporated at various points in the system validated the integrity of the data as items are imported into the system, assigned or reassigned to groups, or are modified in preparation for the automatic matching function. These control checks, and the accompanying reports that are automatically generated, provide a clear audit trail of the changes that are made to the data. The control checks, and
associated reports, also ensure that the user cannot put the account of balance unintentionally. With the short-term PC system, the user might not know about an out-of-balance condition until the end of the day when the data entry was completed. This situation had occurred often and had seriously delayed the balancing process on each occurrence. The automated matching system also provided many new reporting capabilities allowing the user to select more specific reports.
CHAPTER 5
CONCLUSIONS

This project represents one of the most important projects undertaken by Dominion Bankshares Corporation in 1991. The problems with the FRB account uncovered early in the year led to the realization that the requirement of the account had grown beyond the bounds of a manual process. The process needed to be automated so that the personnel resources could be utilized for the research and resolution of exception items. In hindsight, the situation could be attributed to several factors:

- There was insufficient training for the accounting staff involved in the reconciliation process; areas originating transactions would make changes without informing the accounting department.
- There was not a large enough staff to take on the extra burden caused by the consolidation of the Virginia banks.
- In the Fall of 1991, the clearing of Mortgage-Backed securities (MBS's) was transferred from Chemical Bank to the FRB for cost savings, and the accounting staff was not trained in these very complicated transactions which involved timing differences.
- The managers of the accounting department were not informed of the worsening condition of the account and did not take steps to correct it.

Due to these factors, no standards had been put in place for making entries to the FRB account. The originators of the entries would not take responsibility for the resolution of items they had originated and there were no standards in place for reporting exception items back to the originator. Therefore the list of exception items grew to an unmanageable size and posed a serious financial risk to the bank because
the possibility of ever collecting on misposted items was very small. In addition, the exception items represented loses on interest that could have been made had the funds been collected in a timely manner.

The short-term PC system provided immediate relief to the manual reconcilement process. It permitted a migration from a biweekly balancing to a daily balancing schedule which was a new requirement imposed by the bank’s executive management in response to the problem. The time savings provided by this system led to the eventual resolution of all exception items more than thirty days old, which significantly reduced the bank’s exposure to financial loss and eased the task of maintaining the list of exception items. The system also provided the ability to group transactions into categories to facilitate the clearing of exception items and to allow the reporting of only relevant items back to their originators. These reporting capabilities helped the originating areas to more easily identify and resolve exception items.

The long-term goal of this project was to automate, to the greatest degree possible, the reconcilement process. Since the automated process would emulate the manual process it was necessary to achieve a detailed understanding of the process and to document it for future reference. The manual process was very complex but could be broken down into manageable pieces by segregating the transactions into groups. Each of these groups was studied separately, however some transactions affected more than one group. For each group it was necessary to develop not only an understanding of the original transaction, but also how to incorporate return items, adjustments, and mispostings. The bank’s GL system and the FRB’s IAS system were studied because they are the source of data used in the reconcilement process. The source documents and reports produced by both systems were acquired
in a data file format for use in the automated matching system. Several bank systems were studied because of their ties to the GL system through automated interfaces. Alternatives for acquiring the needed detail information from each system were discussed.

The programming accomplished as part of this project has applications to accounts other than the FRB account. The short-term PC program was implemented for the reconcilement of approximately fifteen other general ledger accounts in Corporate Accounting. Thus this program could be adapted for an unlimited number of account reconciliations throughout the bank. Few other accounts present the complexity of transactions found in the FRB account, but if such accounts were identified, the long-term PC program could be adapted as necessary.

Because the development of the PC-based automated matching program was under very strict time constraints, certain aspects of the system were not pursued to their fullest potential. If the system were more permanent, the following enhancements should be pursued:

- Obtain the Money Market II data file containing detail transactions by security and transaction type.
- Obtain the Cash Letter (CPCS) file containing individual immediate and deferred cash letter amounts.
- Obtain Mortgage-Backed Securities system file containing trade and paydown transactions by security.
- Obtain a detail data file from Dominion Trust Company's Mellon system for security transactions.
- Incorporate Funds Transfer System detail transactions and match at detail level.

Overall, the PC system provides automatic matching of approximately 60 percent of transactions across all balancing groups. Many members of the accounting staff believe that the
DISC system will not surpass this performance because the automated matching program was designed specifically for the FRB account. Certain aspects of the PC system have advantages over those of the DISC system.

- The PC system permits balancing by group which facilitates the identification of out-of-balance conditions.
- It emulates the manual process which eased the conversion process.
- The accounting staff is unfamiliar with the DISC system and extensive training will be required.
- The conversion of exception items from PC to DISC will require manual entry; The conversion to the long-term PC system was accomplished automatically.
- The installation of the DISC system will require many resources from the Systems Development area, and continued support for maintenance and daily processing will be required.
- Further customization of the system will not be possible because DISC is a packaged system.

The greatest benefits of the DISC system are its faster processing speed, which is inherent in a mainframe environment, and it will allow a consolidation of the account reconciliation into a single department and lend consistency to the process. However, the events of 1991 showed us that the FRB account requires special consideration, and will continue to do so. Thus the continued use of a customized system may be justifiable.
APPENDIX A

SAMPLE FRB STATEMENT AND DATA FILE LAYOUT
<table>
<thead>
<tr>
<th>Description</th>
<th>Debit / Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opening Balance As Of 04/26</td>
<td>80,706,399.60</td>
</tr>
<tr>
<td>05039 NATURED DEF CR</td>
<td>630.26</td>
</tr>
<tr>
<td>08041 FOOD COUPONS</td>
<td>5,935,353.06</td>
</tr>
<tr>
<td>08061 TREAS CPD</td>
<td>58,561.00</td>
</tr>
<tr>
<td>08062 TREAS CPD</td>
<td>80,442.00</td>
</tr>
<tr>
<td>08131 TT&amp;L - DIR INVEST</td>
<td>139,003.00</td>
</tr>
<tr>
<td>08132 DIR INVEST</td>
<td>2,975,027.77</td>
</tr>
<tr>
<td>08133 DIR INVEST</td>
<td>180,806,707.22</td>
</tr>
<tr>
<td>08134 DIR INVEST</td>
<td>1,957,858.09</td>
</tr>
<tr>
<td>08135 DIR INVEST</td>
<td>20,038.82</td>
</tr>
<tr>
<td>08136 DIR INVEST</td>
<td>77,824.81</td>
</tr>
<tr>
<td>08137 DIR INVEST</td>
<td>134,075.20</td>
</tr>
<tr>
<td>08138 DIR INVEST</td>
<td>767,448.83</td>
</tr>
<tr>
<td>08139 DIR INVEST</td>
<td>816,286.74</td>
</tr>
<tr>
<td>BR</td>
<td>BATCH REF</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
</tr>
<tr>
<td>6206</td>
<td>(804) 697-8363</td>
</tr>
<tr>
<td></td>
<td>00002</td>
</tr>
<tr>
<td></td>
<td>00002</td>
</tr>
<tr>
<td></td>
<td>00002</td>
</tr>
<tr>
<td></td>
<td>00002</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>(804) 697-8374</td>
</tr>
<tr>
<td></td>
<td>20441</td>
</tr>
<tr>
<td></td>
<td>20461</td>
</tr>
<tr>
<td></td>
<td>20441</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5700</td>
<td>(800) 322-0570</td>
</tr>
<tr>
<td></td>
<td>25618</td>
</tr>
<tr>
<td></td>
<td>25622</td>
</tr>
<tr>
<td></td>
<td>24250</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5700</td>
<td>(800) 322-0570</td>
</tr>
<tr>
<td></td>
<td>22547</td>
</tr>
<tr>
<td></td>
<td>26018</td>
</tr>
<tr>
<td></td>
<td>22547</td>
</tr>
<tr>
<td></td>
<td>23894</td>
</tr>
<tr>
<td></td>
<td>23894</td>
</tr>
<tr>
<td></td>
<td>25645</td>
</tr>
<tr>
<td></td>
<td>25631</td>
</tr>
<tr>
<td></td>
<td>24253</td>
</tr>
<tr>
<td></td>
<td>24253</td>
</tr>
<tr>
<td></td>
<td>25645</td>
</tr>
<tr>
<td></td>
<td>25631</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>(804) 697-8321</td>
</tr>
<tr>
<td></td>
<td>50315</td>
</tr>
<tr>
<td>3000</td>
<td>(804) 697-8321</td>
</tr>
<tr>
<td></td>
<td>50329</td>
</tr>
<tr>
<td>3000</td>
<td>(804) 697-8321</td>
</tr>
<tr>
<td></td>
<td>40139</td>
</tr>
<tr>
<td></td>
<td>40139</td>
</tr>
<tr>
<td></td>
<td>40141</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>BR</td>
<td>BATCH</td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
</tr>
<tr>
<td>3700</td>
<td>(804)</td>
</tr>
<tr>
<td></td>
<td>26001</td>
</tr>
<tr>
<td></td>
<td>26001</td>
</tr>
<tr>
<td></td>
<td>26001</td>
</tr>
<tr>
<td></td>
<td><strong>SUBTOTAL</strong></td>
</tr>
<tr>
<td>3811</td>
<td>(804)</td>
</tr>
<tr>
<td>00002</td>
<td>0038</td>
</tr>
<tr>
<td>00001</td>
<td>0177</td>
</tr>
<tr>
<td>00001</td>
<td>0179</td>
</tr>
<tr>
<td>00001</td>
<td>0178</td>
</tr>
<tr>
<td>00002</td>
<td>2040</td>
</tr>
<tr>
<td>00001</td>
<td>0176</td>
</tr>
<tr>
<td>00001</td>
<td>2113</td>
</tr>
<tr>
<td>00002</td>
<td>2068</td>
</tr>
<tr>
<td></td>
<td><strong>SUBTOTAL</strong></td>
</tr>
<tr>
<td>3300</td>
<td>(804)</td>
</tr>
<tr>
<td>3400</td>
<td>(304)</td>
</tr>
<tr>
<td></td>
<td><strong>SUBTOTAL</strong></td>
</tr>
<tr>
<td>8000</td>
<td>(804)</td>
</tr>
<tr>
<td>00011</td>
<td>0004</td>
</tr>
<tr>
<td>00015</td>
<td>2077</td>
</tr>
<tr>
<td>00013</td>
<td>0001</td>
</tr>
<tr>
<td>00015</td>
<td>0411</td>
</tr>
<tr>
<td>00011</td>
<td>0002</td>
</tr>
<tr>
<td></td>
<td><strong>SUBTOTAL</strong></td>
</tr>
<tr>
<td>8000</td>
<td>(804)</td>
</tr>
<tr>
<td>00022</td>
<td>7330</td>
</tr>
<tr>
<td>00022</td>
<td>4054</td>
</tr>
<tr>
<td>00020</td>
<td>0005</td>
</tr>
<tr>
<td></td>
<td><strong>SUBTOTAL</strong></td>
</tr>
<tr>
<td>6200</td>
<td>(804)</td>
</tr>
<tr>
<td>00001</td>
<td>0110</td>
</tr>
<tr>
<td>00001</td>
<td>0120</td>
</tr>
<tr>
<td></td>
<td><strong>SUBTOTAL</strong></td>
</tr>
<tr>
<td>BR</td>
<td>BATCH REF</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
</tr>
<tr>
<td>6103 (804) 697-8373</td>
<td>70011 SAV BONDS ISSUED</td>
</tr>
<tr>
<td>50015 2313</td>
<td></td>
</tr>
<tr>
<td>50015 2019</td>
<td></td>
</tr>
<tr>
<td>50015 2198</td>
<td></td>
</tr>
<tr>
<td>50015 2225</td>
<td></td>
</tr>
<tr>
<td>50015 7810</td>
<td></td>
</tr>
<tr>
<td>50015 2049</td>
<td></td>
</tr>
<tr>
<td>50015 7232</td>
<td></td>
</tr>
<tr>
<td>50015 7932</td>
<td></td>
</tr>
<tr>
<td>50015 2562</td>
<td></td>
</tr>
<tr>
<td>50015 2202</td>
<td></td>
</tr>
<tr>
<td>50015 7811</td>
<td></td>
</tr>
<tr>
<td>50015 7529</td>
<td></td>
</tr>
<tr>
<td>50015 7532</td>
<td></td>
</tr>
<tr>
<td>50015 7530</td>
<td></td>
</tr>
<tr>
<td>50015 7525</td>
<td></td>
</tr>
<tr>
<td>50015 7512</td>
<td></td>
</tr>
<tr>
<td>50015 7003</td>
<td></td>
</tr>
<tr>
<td>50015 7882</td>
<td></td>
</tr>
<tr>
<td>50015 2097</td>
<td></td>
</tr>
<tr>
<td>50015 7524</td>
<td></td>
</tr>
<tr>
<td>50015 7531</td>
<td></td>
</tr>
<tr>
<td>50015 2042</td>
<td></td>
</tr>
<tr>
<td>50015 2019</td>
<td></td>
</tr>
<tr>
<td>50015 2752</td>
<td></td>
</tr>
<tr>
<td>50015 7521</td>
<td></td>
</tr>
<tr>
<td>50015 7526</td>
<td></td>
</tr>
<tr>
<td>50015 2704</td>
<td></td>
</tr>
<tr>
<td>50015 7527</td>
<td></td>
</tr>
<tr>
<td><strong>SUBTOTAL</strong></td>
<td>28</td>
</tr>
</tbody>
</table>

**OWN ACTIVITY RECAP**

<table>
<thead>
<tr>
<th>RESPONDENT ACTIVITY</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0510-0141-4</td>
<td>4,709,067.62</td>
<td>38,031,482.00</td>
</tr>
<tr>
<td>0514-0066-2</td>
<td>25.00</td>
<td></td>
</tr>
<tr>
<td>0514-0132-8</td>
<td>8,893,455.79</td>
<td>4,582,462.91</td>
</tr>
<tr>
<td>0514-0140-9</td>
<td>9,353.00</td>
<td>659.68</td>
</tr>
<tr>
<td>0514-0156-4</td>
<td>5,045,188.90</td>
<td>843,772.01</td>
</tr>
<tr>
<td>0514-0167-4</td>
<td>10,737.89</td>
<td>13,817.89</td>
</tr>
<tr>
<td>0514-0283-5</td>
<td>556,009.93</td>
<td>14,769.44</td>
</tr>
<tr>
<td>0514-0375-6</td>
<td>416,453.99</td>
<td>43,550.78</td>
</tr>
<tr>
<td>0514-0389-2</td>
<td>18,964.71</td>
<td>5,721.72</td>
</tr>
<tr>
<td>0514-0399-9</td>
<td>11,053.69</td>
<td></td>
</tr>
<tr>
<td>0514-0416-3</td>
<td>4,626.58</td>
<td></td>
</tr>
<tr>
<td>0514-0460-0</td>
<td>29,512.79</td>
<td>13,335.92</td>
</tr>
<tr>
<td>REF</td>
<td>5,000.00</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>514-0500-7</td>
<td>100,000.00</td>
<td></td>
</tr>
<tr>
<td>514-0515-7</td>
<td>8,524.32</td>
<td></td>
</tr>
<tr>
<td>515-0305-1</td>
<td>2,360.34</td>
<td></td>
</tr>
<tr>
<td>514-0122-0</td>
<td>1,120,072.20</td>
<td></td>
</tr>
<tr>
<td>550-0320-1</td>
<td>1,565,210.60</td>
<td></td>
</tr>
<tr>
<td>560-0430-3</td>
<td>1,155.26</td>
<td></td>
</tr>
<tr>
<td>560-0760-4</td>
<td>4,524,262.56</td>
<td></td>
</tr>
<tr>
<td>514-7255-5</td>
<td>470,701.40</td>
<td></td>
</tr>
<tr>
<td>514-7308-8</td>
<td>220,175.19</td>
<td></td>
</tr>
<tr>
<td>514-8166-9</td>
<td>682.46</td>
<td></td>
</tr>
<tr>
<td>514-8194-4</td>
<td>13,430.26</td>
<td></td>
</tr>
<tr>
<td>514-8199-9</td>
<td>7,422.62</td>
<td></td>
</tr>
<tr>
<td>514-8288-6</td>
<td>573.00</td>
<td></td>
</tr>
<tr>
<td>514-8342-1</td>
<td>6,930.95</td>
<td></td>
</tr>
<tr>
<td>SUBTOTAL</td>
<td>37,868,848.32</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>289,457,614.02</td>
<td></td>
</tr>
</tbody>
</table>

**Opening Balance**: 80,706,399.60

**Total Debits**: 289,457,614.02

**Total Credits**: 288,545,880.20

**Closing Balance as of 04/26**: 79,794,665.78

---

**Statement of Deferred Activity**

**Available 04/29**

<table>
<thead>
<tr>
<th>REF</th>
<th>53,220.93</th>
</tr>
</thead>
<tbody>
<tr>
<td>697-8058</td>
<td>346,383.38</td>
</tr>
<tr>
<td>697-8035</td>
<td>625,948.80</td>
</tr>
<tr>
<td>697-8035</td>
<td>883,972.23</td>
</tr>
<tr>
<td>SUBTOTAL</td>
<td>1,909,525.34</td>
</tr>
</tbody>
</table>

---

**Own Activity Recap**

| 10 | 5,483,775.95 |

**Respondent Activity**
FEDERAL RESERVE BANK OF RICHMOND
IAS DEPOSIT ACCOUNTING
DFI RECONCILEMENT TAPE FORMAT

DETAIL RECORD

<table>
<thead>
<tr>
<th>STARTING POSITION</th>
<th>PICTURE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X(01)</td>
<td>RECORD TYPE (D=DETAIL)</td>
</tr>
<tr>
<td>2</td>
<td>X(01)</td>
<td>FILLER (SPACE)</td>
</tr>
<tr>
<td>3</td>
<td>X(01)</td>
<td>STATEMENT SECTION /1</td>
</tr>
<tr>
<td>4</td>
<td>X(02)</td>
<td>FILLER (SPACE)</td>
</tr>
<tr>
<td>6</td>
<td>X(05)</td>
<td>TRANSACTION CODE</td>
</tr>
<tr>
<td>11</td>
<td>9(13)V99</td>
<td>AMOUNT</td>
</tr>
<tr>
<td>26</td>
<td>X(01)</td>
<td>DR/CR INDICATOR (D OR C)</td>
</tr>
<tr>
<td>27</td>
<td>X(09)</td>
<td>RESPONDENT TRANSIT ROUTING NUMBER</td>
</tr>
<tr>
<td>36</td>
<td>X(04)</td>
<td>BRANCH NUMBER OF ACCOUNT HOLDER</td>
</tr>
<tr>
<td>40</td>
<td>X(09)</td>
<td>TIERED RESPONDENT NUMBER</td>
</tr>
<tr>
<td>49</td>
<td>X(09)</td>
<td>OFFSET TRANSIT ROUTING NUMBER</td>
</tr>
<tr>
<td>58</td>
<td>X(09)</td>
<td>BATCH NUMBER</td>
</tr>
<tr>
<td>67</td>
<td>X(04)</td>
<td>REFERENCE NUMBER</td>
</tr>
<tr>
<td>71</td>
<td>X(06)</td>
<td>PROCESS DATE (YYMMDD)</td>
</tr>
<tr>
<td>77</td>
<td>X(06)</td>
<td>AVAILABLE DATE (YYMMDD)</td>
</tr>
<tr>
<td>83</td>
<td>X(09)</td>
<td>CONTROL ITEM</td>
</tr>
<tr>
<td>92</td>
<td>X(04)</td>
<td>RSDO CONTROL ITEM CODE (NOT USED)</td>
</tr>
<tr>
<td>96</td>
<td>X(15)</td>
<td>RSDO CONTROL ITEM VALUE (NOT USED)</td>
</tr>
<tr>
<td>111</td>
<td>X(20)</td>
<td>RESERVED FOR FUTURE ACCOUNTING USE</td>
</tr>
<tr>
<td>131</td>
<td>X(02)</td>
<td>APPLICATION ID (VALUE OF &quot;FT&quot;)</td>
</tr>
<tr>
<td>133</td>
<td>X(04)</td>
<td>TYPE/SUBTYPE</td>
</tr>
<tr>
<td>137</td>
<td>X(16)</td>
<td>REFERENCE NUMBER</td>
</tr>
<tr>
<td>153</td>
<td>X(26)</td>
<td>IMAD NUMBER</td>
</tr>
<tr>
<td>179</td>
<td>X(22)</td>
<td>FILLER</td>
</tr>
</tbody>
</table>

1/ I = DEFERRED AVAILABLE TODAY AND OTHER IMMEDIATES
   D = DEFERRED PROCESSED TODAY BY AVAILABLE DATE
   F = FUTURE PROCESSED TODAY BY AVAILABLE DATE
<table>
<thead>
<tr>
<th>Account Code</th>
<th>Description</th>
<th>Source Code</th>
<th>Effective Date</th>
<th>AMOUNT</th>
<th>PAY DATE</th>
<th>NOTE</th>
<th>PAY CODE</th>
<th>PAY CODE</th>
<th>EFFECTIVE DATE</th>
<th>PAY CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000400413</td>
<td>EP SETTLEMENT CHARGES</td>
<td>0000000013</td>
<td>0901</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000400413</td>
<td>EP SETTLEMENT CHARGES</td>
<td>0000000013</td>
<td>0901</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

PROGRAMMER'S DOCUMENTATION
DOMINION BANKSHARES CORPORATION

PROGRAMMER'S DOCUMENTATION

GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

DECEMBER, 1991

CONTENTS

1. PROGRAM ABSTRACT
2. FLOW CHART
3. MODULE CHART
4. MODULE DESCRIPTIONS
5. FILE LAYOUTS
6. TEST SCRIPT
7. PROGRAM LISTING

DEPARTMENT: Corporate Accounting
LANGUAGE: SMART II
PROGRAMMER: Shea T. Chaney, Strategic Analysis
PROGRAM DESCRIPTION

DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM ABSTRACT

LANGUAGE: SMARTII

DATE: December 3, 1991

PROGRAMMER: Shea Chaney

DEPARTMENT: Strategic Analysis
PROGRAM DESCRIPTION

DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

ABSTRACT DESCRIPTION

The purpose of this program is to automate the reconciliation and balancing of Dominion Bank's general ledger (GL) account 1060, which is the Federal Reserve Bank (FRB) cash account. The GL and FRB statements of account are available in data files and contain the same information as the actual printed reports. These files are read into separate data base files representing the GL, or "We" side, and the FRB, or "They" side, of the reconciliation process. Within each file, transactions are identified as debits or credits, with "We Debits" matching "They Credits," and "They Debits" matching "We Credits."

The balancing process is accomplished most efficiently by breaking the transactions down into groups. These groups correspond to the types of transactions that are cleared through account 1060. The ten balancing groups are identified below:

CB - Correspondent Banks
ACH - Automated Clearing House
CC - Coin & Currency Orders
CL - Cash Letters
CPN - Bond Coupon Collections
FS - Food Stamp Collections
FTS - Funds Transfer System - Wire Activity
INV - Security Investments
OTH - Various Fees and Unidentified Transactions
SB - U.S. Savings Bond Sales

Items are reported to their originating area once the items reach a certain age. For some groups, items must be reported to different originating areas. Therefore items are further categorized into report groups as follows:
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

ABSTRACT DESCRIPTION

<table>
<thead>
<tr>
<th>GROUP</th>
<th>REPORT GROUP</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td>CBACH</td>
<td>Correspondent Bank - ACH</td>
</tr>
<tr>
<td></td>
<td>CBCL</td>
<td>Correspondent Bank - Cash Letters</td>
</tr>
<tr>
<td></td>
<td>CBOTH</td>
<td>Correspondent Bank - Other</td>
</tr>
<tr>
<td>ACH</td>
<td>EACH</td>
<td>Eastern VA Op Center ACH</td>
</tr>
<tr>
<td></td>
<td>WACH</td>
<td>Western VA OP Center ACH</td>
</tr>
<tr>
<td>CC</td>
<td>CCGH</td>
<td>Coin &amp; Currency Gr. Ham. Roads</td>
</tr>
<tr>
<td></td>
<td>CCNA</td>
<td>Coin &amp; Currency Roanoke</td>
</tr>
<tr>
<td></td>
<td>CCNV</td>
<td>Coin &amp; Currency Northern Virginia</td>
</tr>
<tr>
<td></td>
<td>CCR</td>
<td>Coin &amp; Currency Richmond</td>
</tr>
<tr>
<td></td>
<td>CCSV</td>
<td>Coin &amp; Currency Shen. Valley</td>
</tr>
<tr>
<td>CL</td>
<td>ECL</td>
<td>EVOC Cash Letters</td>
</tr>
<tr>
<td></td>
<td>WCL</td>
<td>WVOC Cash Letters</td>
</tr>
<tr>
<td></td>
<td>ERET</td>
<td>EVOC Returns</td>
</tr>
<tr>
<td></td>
<td>WRET</td>
<td>WVOC Returns</td>
</tr>
<tr>
<td>CPN</td>
<td>WVCPN</td>
<td>WVOC Coupons</td>
</tr>
<tr>
<td></td>
<td>TRCPN</td>
<td>Trust Coupons</td>
</tr>
<tr>
<td>FS</td>
<td>EVOC</td>
<td>EVOC Food Stamps</td>
</tr>
<tr>
<td></td>
<td>WVOC</td>
<td>WVOC Food Stamps</td>
</tr>
<tr>
<td>INV</td>
<td>MM</td>
<td>Money Market II</td>
</tr>
<tr>
<td></td>
<td>AIM</td>
<td>Aim Portfolio</td>
</tr>
<tr>
<td></td>
<td>MBS</td>
<td>MBS Trades</td>
</tr>
<tr>
<td></td>
<td>MBSPD</td>
<td>MBS Paydowns</td>
</tr>
<tr>
<td></td>
<td>TRUST</td>
<td>Trust Company</td>
</tr>
<tr>
<td></td>
<td>OTH</td>
<td>Unidentified Investments</td>
</tr>
</tbody>
</table>

The FRB uses approximately 400 5-character numeric transaction codes to describe the transactions that are reported on our statement of account. The GL and FRB
transactions can be classified into ten balancing groups based on the transaction code. This permits an automated matching process where the GL and FRB files can be compared and the resulting matched and unmatched items can be reported. Using the amounts associated with matched and unmatched items, the account can be balanced on a daily basis.
On a daily basis the GL and FRB files are downloaded from the bank's mainframe computer and imported into their respective files. Each file is broken out into balancing group files based on the transaction code. The user is given the opportunity to assign a group code to any items that could not be automatically assigned. Following the assignment function, the daily and cumulative outstanding items for a group are merged together. The user can then modify items within a group by altering descriptions or group code. The user also has the ability to replace one item with several items, or to replace several items with one item. At several points in the assignment, merge, and modification function, control checks are made to ensure that no data is lost and that the daily net activity is maintained.

At this point, the GL and FRB transactions within a balancing group can be automatically compared. Matching items are reported to the user and marked appropriately. The user can print a report of the outstanding items within a group at any point in the process. These reports can be used to pair-off items manually and then the information can be entered into each group file. Paired-off items are deleted from a file only if the activity nets to zero. This ensures that the account will never fall out of balance. Various types of reports are available for reporting outstanding and stale items.

The following chart shows the flow of the program. Each functional piece of the system was developed as a separate program, and the movement from program to program is facilitated by various menus.
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME : GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

FLOW CHART
PROGRAM NAME: Settlement Program

1. Import Daily G/L Activity Report from FIX or Import FRB Daily Statement

2. Calculate net activity for the entire file

3. Segregate file into identified transaction group utilizing a sort table

4. Calculate net activity by group and by total. Compare to net activity total above.

5. Does net total activity agree? No: Generate error report; identify and resolve differences. Yes:

6.
Generate a report of unassigned transactions.

Manually assign reported transactions to the appropriate group.

Calculate net activity by group and total file. Compare group activity to net total.

Do net total amounts agree?

Modify file by group to facilitate matching process.

Recalculate net activity by group and compare to pre-modification.

Update identified items file.

Generate error report; identify and resolve differences.
amounts.

Do not amounts agree?

No

Generate error report; identify and resolve differences.

Yes

Perform automated matching, create matched items file, check debits and credits

Do debits equal credits?

No

Mark items as matched in status field; leave items in cumulative file

Yes

Generate matched items report; remove records from cumulative file

Generate a report of outstanding items by group, including any unbalanced matches

Clearance process on outstanding items; assign detailed group codes
Perform control check to ensure debits = credits by group

Do debits equal credits?

- Yes
  - Print report of cleared items; remove from cumulative file
  - Print report of outstanding items by detailed group
  - Generate settlement sheet

- No
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE CHART
PROGRAM DESCRIPTION

DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE DESCRIPTIONS
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME : GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE DESCRIPTION - IMPORT FILES

This module performs the file import function and splits records from GL and FRB into separate files based on balancing group. The GLFILE and FRBFILE (data files) previously downloaded from the mainframe to diskette using Arbiter are read into their respective database files GL.DB and FRB.DB. File layouts for these files are included in a later section.

The GL file is imported first and calculations are performed for various fields. The file is broken into group files CBG.DB, ACHG.DB, CCG.DB, CLG.DB, CPNG.DB, FSG,DB, FTSG.DB, INVG.DB, SBG,DB, OTHERG,DB and UNIDG,DB based on the group code calculated for each item. A file layout for the file CBG.DB is included in this document, and all other group files use this same file layout. The net activity from the GL file is checked against the sum of all net activity in each group file to ensure that no records were lost. If the net GL activity is verified, a report of unidentified items is printed and the program jumps to the GL assignment function for unidentified items. If the net GL activity cannot be verified, an error report is printed.

Next, the FRB file is imported and calculations are performed for various fields. The file is then broken into group files CBF.DB, ACHF.DB, CCF.DB, CLF.DB, CPNF.DB, FSF,DB, FTSF.DB, INVF.DB SFG.DB, OTHERF,DB and UNIDF,DB based on the group code calculated for each item. A file layout for the file CBF.DB is included in this document and all other group files use this same file layout. The net activity from the FRB file is checked against the sum of all net activity in each group file to ensure that no records were lost. If the net FRB activity is verified, a report of unidentified items is printed and the program jumps to the FRB assignment function for unidentified items. If the net FRB activity cannot be verified then an error report is printed.
In this module the user is permitted to assign the appropriate group code to transactions that were not automatically coded. During the import module, the records in each file are assigned a balancing group code (CB, ACH, CC, CL, CPN, FS, FTS, INV, OTHER, SB) based on the FRB transaction code. If a GL entry is made with an improper FRB code, a group code is not assigned and the transaction is sent to a file with other unidentified GL transactions. If a transaction in the FRB data file contains a new transaction code, it would be sent to a file with other unidentified FRB transactions.

The GL and FRB files containing unidentified transactions are loaded automatically during the import function, and a group code must be assigned to each item. If a group code is not assigned to an item, it is automatically assigned to group "OTHER."

After the group codes are entered, the file is searched, and each record is sent to its appropriate group file. A control check is made to verify that the net activity in all daily group files sum to the net activity for that day. If this is verified then the daily group files are merged with the cumulative outstanding items for each group. A second control check is made to ensure that the merger of the daily and cumulative group files matches the expected sum of these two files.
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE DESCRIPTION - MODIFY GROUP FILES

In this module the user can modify each balancing group file for the "We" or "They" side. During the first stage of modifications the user can change the transaction code, group code, or report group code for an item.

Once the user has completed these changes, replacements may be made. If the user chooses to make replacements the file is loaded again and the user must enter an "R" in the first column signifying that the item is to be replaced. Then a new screen appears with this item at the top of the file. The user should enter the item(s) that will replace this item. A control check verifies that the replacing item(s) sum to the amount of the replaced item. If the amounts do not match, the user is returned to the replacement screen. The user will not be able to leave this function until the sum of the replacing items is equal to the sum of the replaced items. Replacements can be done individually or in groups.

Following the replacement session the file is searched to determine if any items have been reassigned to different balancing groups. If items have been reassigned, they are sent to the appropriate group file. At this point a control check is made to determine if the net of all group activity is equal to the net of cumulative activity across all groups. A report is printed, regardless of the type of changes that were made, showing any changes in group activity and any change in overall net activity.
PROGRAM DESCRIPTION

DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME : GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE DESCRIPTION - AUTOMATED MATCHING BY GROUP

This program compares the GL and FRB outstanding items within a balancing group, and attempts to "pair-off" matching items. From the match menu (MENUMTCH) the user selects the group desired for automated matching. The program loads the file containing the GL and FRB outstanding items for the group selected, and attempts to "pair-off" matching items. The matching criteria used as the basis for automated matching varies by balancing group. All groups use the debit/credit indicator, FRB transaction code, and amount as matching criteria. Several groups use a fourth matching criteria, Description 2, which contains different information for each of these groups. The table at the end of this section lists the fields used as matching criteria for each balancing group.

When items are matched, a code consisting of "A" followed by "mmdd" (month and day matching occurred) is placed in the MATCH field of an item. A report of the matched items is automatically generated, sorting on ascending amount. If no matches are found, a message prints on the screen.

Matched items are not automatically deleted from the files. It is only by proceeding to the clearing function that these items can be deleted. During the clearing function, a check is made to ensure that the items being deleted net to zero. This ensures that the account will not fall of balance.
### PROGRAM DESCRIPTION

**DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM**

**PROGRAM NAME:** GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

---

**MODULE DESCRIPTION - AUTOMATED MATCHING BY GROUP**

<table>
<thead>
<tr>
<th>GROUP</th>
<th>FIELDS</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>ACH</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>[DEBCRE], [DESC1], [AMOUNT], [D2]</td>
<td>[D2] contains 4-digit Sequence Number</td>
</tr>
<tr>
<td>CL</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>CPN</td>
<td>[DEBCRE], [DESC1], [AMOUNT], [D2]</td>
<td>[D2] contains 4-digit Sequence Number</td>
</tr>
<tr>
<td>FS</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>FTS</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>INV</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>OTHER</td>
<td>[DEBCRE], [DESC1], [AMOUNT]</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>[DEBCRE], [DESC1], [AMOUNT], [D2]</td>
<td>[D2] contains first 4 characters of issuing agent (branch) number</td>
</tr>
</tbody>
</table>

[DEBCRE] = "10" for a GL debit, "60" for a GL credit (FRB transactions are converted to match GL coding)

[DESC1] = 4-character FRB Transaction Code

[AMOUNT] = 16-character numeric field
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME : GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE DESCRIPTION - CLEARING BY GROUP

This program allows the user to clear items from each balancing group. From the clearing function menu (MENCLEAR) the user chooses the desired group from which to clear items. The user is asked if there are items to be cleared from the GL group file; if the response is yes, the file of outstanding GL items for that group is brought to the screen. If the items in this file were previously automatically matched, the assigned matching code ("A" and date) will be found in the MATCH column. A user can "unmatch" items by removing this code from the MATCH field. Additional items can be cleared by adding a code (user initials and number) in the MATCH field.

When the user has finished clearing in the GL group file, the same can be done for the FRB group file. At this point the cleared items from the GL and FRB group files are added to determine if the items net to zero. If the cleared transactions net to zero, a report listing the cleared items is printed and the items are deleted from their respective group files. If the cleared transactions do not net to zero a similar report is printed but the items remain in their respective files.

Items that are intended for clearing must have a code entered in the MATCH field. However, they will only be deleted when the net activity in the GL and FRB files for that group nets to zero. The only field that can be changed by the user during this function is the MATCH field.
This program copies the files containing the cumulative GL and FRB outstanding items for each balancing group to diskettes to provide a copy of the files. The first five groups of files (CB, ACH, CC, CL, CPN) are copied to diskette #1 and the remaining five groups (FS, FTS, INV, OTHER, SB) are copied to diskette #2. The user is prompted to insert the appropriate diskette in drive A:. The user must initiate this backup program from the main menu; the group files are not automatically backed up to diskettes. If the user must restore these files, this can be done using the restore option from the main menu. In the restore function, the user has the option of restoring the files from diskettes or from the hard drive.
PROGRAM DESCRIPTION

DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME : GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

MODULE DESCRIPTION - RESTORE PRIOR DAY'S FILES

This program permits the user to "undo" any work done for the current day. This is accomplished by replacing the current files of outstanding items with the files backed up during the previous balancing cycle. Each time the import function is run, the files containing outstanding GL and FRB items for each balancing group are backed up on the PC hard drive in subdirectory c:\AUTO\RESTORE. In addition, these files can be backed up to diskettes on demand using the backup function of the main menu. The restore program gives the user the option of restoring files from the hard drive or from diskettes. The files stored on the hard drive contain the outstanding items before the items for the current day were imported and assigned to groups. The files stored on diskettes contain the most recent files that were backed up using the backup function from the main menu.
These programs produce a listing of outstanding items for the selected balancing group with an effective date falling within the selected date range. The first screen prompts the user for the desired group code, as-of date, and date range for the report header. The second screen requires that the user modify a query with the start and end dates of the desired date range. If records are found fitting the date and group criteria, then the report prints, otherwise the user is notified that no records were found matching this criteria.
<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-1.000</td>
<td></td>
</tr>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-1.000</td>
<td></td>
</tr>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-5.000</td>
<td></td>
</tr>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-0.750</td>
<td></td>
</tr>
</tbody>
</table>

Total: -26.018.00
Count: 0

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>3.000</td>
<td></td>
</tr>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>3.000</td>
<td></td>
</tr>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.385</td>
<td></td>
</tr>
<tr>
<td>CCOH</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

Total: 35.000.00
Count: 7

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-1.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-1.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.385</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

Total: -3.412.956.00
Count: 12

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>350.00</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>965.000</td>
<td></td>
</tr>
</tbody>
</table>

Total: 965.350.00
Count: 2

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-1.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.385</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

Total: 247.050.00
Count: 9

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

Total: 1.000.00
Count: 1

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Group File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Desc 2</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>-1.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CCNA</td>
<td>PBB</td>
<td>CREDIT</td>
<td>02/28/92</td>
<td>6301</td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

Total: -1.000.00
Count: 4
<table>
<thead>
<tr>
<th>Group</th>
<th>File</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Credit</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>357.00</td>
</tr>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>717.00</td>
</tr>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>17.00</td>
</tr>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>140.00</td>
</tr>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>200.00</td>
</tr>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>100.00</td>
</tr>
<tr>
<td>C00</td>
<td>PRB</td>
<td>DEBIT</td>
<td>02/28/92</td>
<td>6391</td>
<td></td>
<td>100.00</td>
</tr>
</tbody>
</table>

Total: -357.000.00
Count: 5

Grand Total: -2,215,460.00
Count: 54
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

REPORT DESCRIPTION - GL/FRB REPORT GROUP FILE FOR DATE RANGE

These programs produce a listing of outstanding items for the selected report group with an effective date falling within the selected date range. The first screen prompts the user for the desired group code, as-of date, and date range for the report header. The second screen requires that the user modify a query with the start and end dates of the desired date range. If records are found fitting the date and group criteria, then the report prints; otherwise the user is notified that no records were found matching this criteria.
<table>
<thead>
<tr>
<th>Group</th>
<th>Name</th>
<th>File Code</th>
<th>Type</th>
<th>Date</th>
<th>Description</th>
<th>Amount</th>
<th>Match Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>CCBM</td>
<td>GL</td>
<td>DEBIT</td>
<td>02/20/92</td>
<td>7045 FED FRANK SHORT ORE BILL</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6494 COIN GROUNDS</td>
<td></td>
<td>300.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6495 COIN CURRENCY CERTIFICATES</td>
<td></td>
<td>1,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6496 COIN SHIPPED FOR 36234641</td>
<td></td>
<td>1,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6497 COIN CURRENCY</td>
<td></td>
<td>10,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6498 COIN CURRENCY VALLEY HALL</td>
<td></td>
<td>70,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6499 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>510,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6500 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>1,110,700.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Type Total:</strong></td>
<td></td>
<td>1,706,505.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Count 8:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6501 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-1,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6502 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-10,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6503 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-70,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6504 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-500,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Type Total:</strong></td>
<td></td>
<td>-786,300.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Count 9:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6505 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>965,350.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>File Total:</strong></td>
<td></td>
<td>1,422,355.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Count 17:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6506 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-1,500.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6507 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-10,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6508 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-50,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6509 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>-2,000,000.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Type Total:</strong></td>
<td></td>
<td>-2,412,700.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Count 12:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6510 COIN CURRENCY MONEY DOLLAR</td>
<td></td>
<td>965,350.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>File Total:</strong></td>
<td></td>
<td>-2,412,700.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Count 14:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
These programs give the user the ability to print a report of all GL or FRB items that have been replaced, within their respective group files, since the last time that report was printed. There are two database files, MODFILEG and MODFILEF, that accumulate the records involved in replacement for GL and FRB items respectively. The report lists the items in order by balancing group code, and within group by replacement code, which uniquely identifies each set of records involved in a replacement.
<table>
<thead>
<tr>
<th>GROUP</th>
<th>RECLAS</th>
<th>CODE</th>
<th>GROUP CODE</th>
<th>DESCRIPTION</th>
<th>DESCR</th>
<th>AMOUNT</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>850</td>
<td>CCRB</td>
<td>1004</td>
<td></td>
<td>7145</td>
<td>-111,000</td>
<td>02/28/92</td>
</tr>
<tr>
<td>R</td>
<td>850</td>
<td>CCRB</td>
<td>1004</td>
<td></td>
<td>7145</td>
<td>-31,000</td>
<td>02/28/92</td>
</tr>
<tr>
<td>R</td>
<td>850</td>
<td>CCRB</td>
<td>1004</td>
<td></td>
<td>7145</td>
<td>91,000</td>
<td>02/28/92</td>
</tr>
<tr>
<td>R</td>
<td>850</td>
<td>CCRB</td>
<td>1004</td>
<td></td>
<td>7145</td>
<td>60,000</td>
<td>02/28/92</td>
</tr>
<tr>
<td>R</td>
<td>850</td>
<td>CCRB</td>
<td>1004</td>
<td></td>
<td>7145</td>
<td>100,000</td>
<td>02/28/92</td>
</tr>
</tbody>
</table>

Replacement Total: 0.00  
Count 6

Group Total: 0.00  
Count 4

Grand Total: 0.00  
Count 6
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME : GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

REPORT DESCRIPTION - ALL OUTSTANDING ITEMS

This program produces a report of all outstanding transactions sorted by the report group code; the report page-breaks on report group. This report is generated by merging all outstanding items across all groups into one file, and then sorting and subtotalling the items by detailed report group, file and type.
Program Description

DBC GL Account 1060 Automated Matching Program

Program Name: GL Account 1060 Automated Matching Program

Report Description - Outstanding Items by Report Group

This program provides a report listing all outstanding items within a report group. Within a report group, items are sorted and subtotalled by the (We Debit, We Credit, FRB Debit, FRB Credit).
This program produces a report of stale items for the desired report group code. Stale items are those that have been outstanding for more than a certain number of days. The number of days beyond which an items is considered stale is determined by the group to which it belongs. The stale item criteria are as follows:

<table>
<thead>
<tr>
<th>GROUP CODE</th>
<th>REPORT GROUP CODE</th>
<th>NUMBER OF DAYS OUTSTANDING</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td>CBACH,CBCL,CBOTH</td>
<td>7</td>
</tr>
<tr>
<td>ACH</td>
<td>EACH,WACH</td>
<td>2</td>
</tr>
<tr>
<td>CC</td>
<td>CCGH,CCNA,CCNV,CCR,CCSV</td>
<td>7</td>
</tr>
<tr>
<td>CL</td>
<td>ECL,WCL,ERET,WRET</td>
<td>14</td>
</tr>
<tr>
<td>CPN</td>
<td>TRCPN,WVCPN</td>
<td>7</td>
</tr>
<tr>
<td>FS</td>
<td>EVOC,WVOC</td>
<td>14</td>
</tr>
<tr>
<td>FTS</td>
<td>FTS</td>
<td>2</td>
</tr>
<tr>
<td>INV</td>
<td>AIM,MM,TRUST,MBS,MBSPD,OTH</td>
<td>14</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTH</td>
<td>7</td>
</tr>
<tr>
<td>SB</td>
<td>SBGH,SBNA,SBNV,SBR,SBSV</td>
<td>7</td>
</tr>
</tbody>
</table>

The user must specify the desired report group and based on this code, the appropriate GL and FRB group files are loaded. Each file is queried for stale items based on the criteria listed above. Stale items from both files are merged into a new file and the report is generated, sorting and subtotalling items by file and type.
PROGRAM DESCRIPTION

DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

REPORT DESCRIPTION - STALE ITEMS FOR ALL GROUPS

This program produces a report of all stale items sorted and subtotalled by report group, file, and type. The stale item criteria used is the same as that used in the report "Stale Items by Report Group." Each set of group files is queried for stale items based on these criteria. Stale items within a balancing group are temporarily merged and then sorted by report group for the purpose of generating the report. Each group is reported separately, but automatically in succession, and the report page breaks on report group code.
This program generates a cover sheet used to balance the 1060 account. In this program, the net activity for each file (GL and FRB) for each balancing group is calculated, and these numbers are summed to determine the net amount of all outstanding items. The user must input the current GL and FRB account balances. The difference between these balances is calculated and compared to the net amount of all outstanding items. If the account is in balance, these numbers should net to zero.
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

FILE LAYOUTS
View: u:alom\u10877\auto\gl.vws
Date Created: 09/19/91
Last Edited: 01/15/92
Field Count: 19
Record Count: 678
Encrypted: No

Data-file: u:alom\u10877\auto\gl.db
Total Fields: 19
Record Length: 171
Last Add Date: 01/12/92
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format COMPANY</td>
<td>gl</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>ACCOUNT</td>
<td>gl</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>CENTER</td>
<td>gl</td>
<td>A 7</td>
<td>R/W</td>
</tr>
<tr>
<td>SOURCE</td>
<td>gl</td>
<td>A 10</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC_1</td>
<td>gl</td>
<td>A 30</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC_2</td>
<td>gl</td>
<td>A 12</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>gl</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>Key:AMOUNT</td>
<td>gl</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Entry Date    | D 8             | R/W D1      |
| Key:EFF_DATE  | gl              |             |

| ENTRY_DATE    | D 8             | R/W D1      |
| DEBCRE        | gl              | A 2         | R/W          |
| DESC1         | gl              | A 4         | R/W          |
| Calc: left([DESC_1],4) |

| DESC1T        | gl              | A 26        | R/W          |
| Calc: right([DESC_1],26) |

| DESC2         | gl              | A 12        | R/W          |
| Calc: [DESC_2]|

| FILE          | gl              | A 3         | R/W          |
| Calc: "GL"    | gl              |             |              |

| WORDS         | gl              | A 6         | R/W          |
| Calc: if [DEBCRE]="10" then "DEBIT" else "CREDIT" |

| SIGN          | gl              | N 8         | R/W 2R       |
| Calc: if [DEBCRE]="10" then [AMOUNT] else -1*[AMOUNT] |

| GROUP         | gl              | A 5         | R/W          |
| Key:GROUP     | gl              |             |              |
| Calc: if (right(left([DESC1T],5),4)="0283" or right(left([DESC1T],5),4)="0376" or |
right(left([DESC1], 5), 4) = "0416" or right(left([DESC1], 5), 4) = "0460" or right(left([DESC1], 5), 4) = "7255" or right(left([DESC1], 5), 4) = "7308" or right(left([DESC1], 5), 4) = "8288" or right(left([DESC1], 5), 4) = "8135" or right(left([DESC1], 5), 4) = "8194" or right(left([DESC1], 5), 4) = "8342" or right(left([DESC1], 5), 4) = "8199") then "CB"

else if ([DESC1] = "5701" or [DESC1] = "5703" or [DESC1] = "5704") then "ACH"

else if ([DESC1] = "6301" or [DESC1] = "6304" or [DESC1] = "6305") then "CC"

else if ([DESC1] = "0503" or [DESC1] = "1019" or [DESC1] = "1501" or [DESC1] = "1502" or [DESC1] = "1515" or [DESC1] = "1528" or [DESC1] = "1531" or [DESC1] = "1533" or [DESC1] = "3005" or [DESC1] = "3008" or [DESC1] = "3010" or [DESC1] = "3016") then "CL"

else if ([DESC1] = "4001") then "CPN"

then "FS"

else if ([DESC1] = "0804" or [DESC1] = "0806" or [DESC1] = "0808") then "FS"

else if ([DESC1] = "1001" or [DESC1] = "1002") then "FTS"

else if ([DESC1] = "2001" or [DESC1] = "2004" or [DESC1] = "2011" or [DESC1] = "2019" or [DESC1] = "2020" or [DESC1] = "2701" or [DESC1] = "2702" or [DESC1] = "2704" or [DESC1] = "4007") then "INV"

else if ([DESC1] = "0802" or [DESC1] = "0811" or [DESC1] = "0812" or [DESC1] = "0813" or [DESC1] = "8403" or [DESC1] = "0818" or [DESC1] = "0815" or [DESC1] = "0822" or [DESC1] = "0825") then "OTHER"

else if ([DESC1] = "7001" or [DESC1] = "7004") then "SB"

else if ([DESC1] = "FTS") then "FTS"

else "??????"

RGROUP

Calc: if [GROUP] = "CB" and

([DESC1] = "0503" or [DESC1] = "1019" or [DESC1] = "1501" or [DESC1] = "1502" or [DESC1] = "1515" or [DESC1] = "1528" or [DESC1] = "1531" or [DESC1] = "1533" or [DESC1] = "3005" or [DESC1] = "3008" or [DESC1] = "3010" or [DESC1] = "3016") then "CBCL"

else if [GROUP] = "CB" and

([DESC1] = "5701" or [DESC1] = "5703" or [DESC1] = "5704") then "CBACH"

else if [GROUP] = "CB" and
<table>
<thead>
<tr>
<th>MATCH</th>
<th>gl</th>
<th>A</th>
<th>S</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;CBOOTH&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>else null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

163

([DESC1]<>"0503" and [DESC1]<>"1019" and [DESC1]<>"1501" and
[DESC1]<>"1502" and [DESC1]<>"1515" and [DESC1]<>"1528" and
[DESC1]<>"1531" and [DESC1]<>"1533" and [DESC1]<>"3005" and
[DESC1]<>"3008" and [DESC1]<>"3010" and [DESC1]<>"3016" and
[DESC1]<>"5701" and [DESC1]<>"5703" and [DESC1]<>"5704") then
### Field Display

<table>
<thead>
<tr>
<th>Format</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABA</td>
<td>FRB</td>
<td>A</td>
<td>9</td>
</tr>
<tr>
<td>STMT_SEC</td>
<td>FRB</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>TRANS_CODE</td>
<td>FRB</td>
<td>A</td>
<td>5</td>
</tr>
</tbody>
</table>

**Key:** TRANS_CODE

<table>
<thead>
<tr>
<th>DC_CODE</th>
<th>FRB</th>
<th>A</th>
<th>1</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMOUNT</td>
<td>FRB</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>REF</td>
<td>FRB</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
</tr>
<tr>
<td>PRO_DATE</td>
<td>FRB</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>AYA_DATE</td>
<td>FRB</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>CON_ITEM</td>
<td>FRB</td>
<td>A</td>
<td>9</td>
<td>R/W</td>
</tr>
<tr>
<td>BANK_DFI</td>
<td>FRB</td>
<td>A</td>
<td>9</td>
<td>R/W</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>FRB</td>
<td>A</td>
<td>2</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Calc:**
- if [DC_CODE]="D" then "60"
- else "00"

<table>
<thead>
<tr>
<th>DESC1</th>
<th>FRB</th>
<th>A</th>
<th>4</th>
<th>R/W</th>
</tr>
</thead>
</table>

**Calc:** left([TRANS_CODE],4)

<table>
<thead>
<tr>
<th>DESC1</th>
<th>FRB</th>
<th>A</th>
<th>26</th>
<th>R/W</th>
</tr>
</thead>
</table>

**Calc:**
- if [ABA]="051402835" then "0283 BANK OF FLOYD"
- else if [ABA]="051403766" then "0376 GRAYSON NATL"
- else if [ABA]="051404163" then "0416 LEE BANK & TRUST"
- else if [ABA]="051404600" then "0460 TAZEWELL"
- else if [ABA]="251472555" then "7255 BEDFORD"
- else if [ABA]="251473088" then "7308 SOUTHWEST VA"
- else if [ABA]="251482888" then "8288 CARILION"
- else if [ABA]="251481355" then "8135 NEWPORT NEWS"
- else if [ABA]="251481944" then "8194 NAVAL HOSP"
- else if [ABA]="251483421" then "8342 YORKTOWN"
else if [ABA]="251481999" then "8199 PFD FIREFIGHTERS"
else [DESC1]

DESC2
FRB
Calc: if [DESC1]="1001" then [BANK_DFI]
else [REF]

FILE
FRB
Calc: "FRB"

WORDS
FRB
Calc: if [DC_CODE]="D" then "DEBIT" else "CREDIT"

SIGN
FRB
Calc: if [DC_CODE]="D" then [AMOUNT] else -1*[AMOUNT]

GROUP
FRB

Key: GROUP
Len: 5  Type: A  Dupes: Yes
ABA
Len: 9  Type: A  Dupes: Yes
BANK_DFI
Len: 9  Type: A  Dupes: Yes

Calc: if ([ABA] = "051402835" or [ABA]="051403766" or
[ABA]="051404163" or
[ABA] = "051404600" or [ABA]="251472555" or
[ABA]="251473088" or
[ABA] = "251482868" or [ABA]="251481355" or
[ABA]="251481944" or
[ABA] = "251483421" or [ABA]="251481999") then "CB"
else if left([DESC1],2)="57" then "ACH"
else if left([DESC1],2)="63" then "CC"
else if ([DESC1]="1011" or [DESC1]="1012" or [DESC1]="1019" or
[DESC1]="1020" or [DESC1]="0502" or [DESC1]="0503" or
left([DESC1],2)="15" or
left([DESC1],2)="30") then "CL"
else if ([DESC1]="4001" or [DESC1]="4002") then "CPN"
else if ([DESC1]="0804" or [DESC1]="0806" or [DESC1]="0808")
then "FS"
else if ([DESC1]="1001" or [DESC2]="1002" or [DESC1]="1099")
then "FTS"
else if ([DESC1]="4003" or [DESC1]="4007" or [DESC1]="4099" or
left([DESC1],2)="20" or
left([DESC1],2)="27" or
left([DESC1],2)="66" or
left([DESC1],2)="82")
then "INV"
else if left([DESC1],2)="70" then "SB"
else if ([DESC1]="0000" or [DESC1]="0501" or [DESC1]="0504" or
[DESC1]="4004" or [DESC1]="4005" or [DESC1]="4006" or
else "??????"

RGROUP  FRB  A  5  R/W
Calc: if [GROUP] = "CB" then
   if ([DESC1]="1011" or [DESC1]="1012" or [DESC1]="1019" or
      [DESC1]="1020" or
      [DESC1]="1011" or [DESC1]="0502" or [DESC1]="0503" or
      left([DESC1],2)="15" or
      left([DESC1],2)="30") then "CBCL"
   else if [GROUP]="CB" and
      (left([DESC1],2)="57") then "CBACH"
   else "CBOTH"
else if [GROUP]="CC" then
   if ([ABA]="051001414" or [ABA]="051401674" or
      [ABA]="051403892") then "CCR"
   else if ([ABA]="051400662" or [ABA]="051401328" or
            [ABA]="051403999") then "CCGH"
   else if ([ABA]="051401409" or [ABA]="051401564" or
            [ABA]="051403520") then "CCSV"
   else if ([ABA]="054001220" or [ABA]="055003201" or
            [ABA]="056004393" or [ABA]="056007604") then
"CCNV"
else "CCNA"
else if [GROUP]="SB" then
   if ([ABA]="051001414" or [ABA]="051401674" or
      [ABA]="051403892") then "SBR"
   else if ([ABA]="051400662" or [ABA]="051401328" or
            [ABA]="051403999") then "SBGH"
   else if ([ABA]="051401409" or [ABA]="051401564" or
            [ABA]="051403520") then "SBSV"
   else if ([ABA]="054001220" or [ABA]="055003201" or
            [ABA]="056004393" or [ABA]="056007604") then
"SBNV"
else "SBNA"
else if [GROUP]="ACH" then
   if ([ABA]="051001414" or [ABA]="051400662" or
      [ABA]="051401328" or [ABA]="051401409"
    or [ABA]="051401564" or [ABA]="051401674"
    or [ABA]="051403520" or [ABA]="051403892"
    or [ABA]="051403999" or [ABA]="054001220"
    or [ABA]="055003201" or [ABA]="056004393"
    or [ABA]="056007604") then "EACH"
else "WACH"
else if [GROUP]="ECL" then
   if ([ABA]="051001414" or [ABA]="051400662" or
      [ABA]="051401328" or [ABA]="051401409"
    or [ABA]="051401564" or [ABA]="051401674"
    or [ABA]="051403520" or [ABA]="051403892"
    or [ABA]="051403999" or [ABA]="054001220"
    or [ABA]="055003201" or [ABA]="055004393"
    or [ABA]="056007604") then "ECL"
else "WCL"
else if [GROUP]="FTS" then "FTS"
else if [GROUP]="OTHER" then "OTH"
else null
<table>
<thead>
<tr>
<th>Field</th>
<th>Display</th>
<th>Data-file Table</th>
<th>Input</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Mask</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RGROUP</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>FILE</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>3</td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>6</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>4</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1T</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>26</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC2</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>12</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>ENTRY_DATE</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>2</td>
<td>R/W</td>
</tr>
<tr>
<td>SIGN</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>MATCH</td>
<td>GLGROUP</td>
<td></td>
<td></td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>Field</td>
<td>Data-file Table</td>
<td>Data</td>
<td>Length</td>
<td>Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------</td>
<td>------</td>
<td>--------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>Input</td>
<td>Type</td>
<td>Status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td>Mask</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>GLGROUP</td>
<td>A 5</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NGROUP</td>
<td>GLGROUP</td>
<td>A 5</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>GLGROUP</td>
<td>A 3</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WORDS</td>
<td>GLGROUP</td>
<td>A 6</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCl</td>
<td>GLGROUP</td>
<td>A 4</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESClT</td>
<td>GLGROUP</td>
<td>A 26</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCl2</td>
<td>GLGROUP</td>
<td>A 12</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMOUNT</td>
<td>GLGROUP</td>
<td>N 8</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFF_DATE</td>
<td>GLGROUP</td>
<td>D 8</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENTRY_DATE</td>
<td>GLGROUP</td>
<td>D 8</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBCRE</td>
<td>GLGROUP</td>
<td>A 2</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td>GLGROUP</td>
<td>N 8</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCH</td>
<td>GLGROUP</td>
<td>A 5</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
View: u:\alm\u18877\auto\cbgc.vws
Date Created: 11/12/91
Last Edited: 11/22/91
Field Count: 15
Record Count: 4
Encrypted: No

Data-file: u:\alm\u18877\auto\CBGC.db
Total Fields: 15
Record Length: 113
Last Add Date: 12/12/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format Mask</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>CBGC</td>
<td>A 1</td>
<td>R/W</td>
</tr>
<tr>
<td>GROUP</td>
<td>CBGC</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>RGROUP</td>
<td>CBGC</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>Key: RGROUP</td>
<td></td>
<td>Len: 5</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>FILE</td>
<td></td>
<td>Len: 3</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>WORDS</td>
<td></td>
<td>Len: 6</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td></td>
<td>Len: 8</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>AMOUNT</td>
<td></td>
<td>Len: 8</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>FILE</td>
<td>CBGC</td>
<td>A 3</td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td>CBGC</td>
<td>A 6</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1</td>
<td>CBGC</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1T</td>
<td>CBGC</td>
<td>A 26</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>CBGC</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>Key: AMOUNT</td>
<td></td>
<td>Len: 8</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>RGROUP</td>
<td></td>
<td>Len: 5</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>DEBCRE</td>
<td></td>
<td>Len: 2</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>DESCI</td>
<td></td>
<td>Len: 4</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td></td>
<td>Len: 8</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>DESC2</td>
<td>CBGC</td>
<td>A 12</td>
<td>R/W</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td>CBGC</td>
<td>D 8</td>
<td>R/W D1</td>
</tr>
<tr>
<td>ENTRY_DATE</td>
<td>CBGC</td>
<td>D 8</td>
<td>R/W D1</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>CBGC</td>
<td>A 2</td>
<td>R/W</td>
</tr>
<tr>
<td>Key: DEBCRE</td>
<td></td>
<td>Len: 2</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>DESCI</td>
<td></td>
<td>Len: 4</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>AMOUNT</td>
<td></td>
<td>Len: 8</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td></td>
<td>Len: 8</td>
<td>Type:A Dupes:Yes</td>
</tr>
<tr>
<td>SIGN</td>
<td>CBGC</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td></td>
<td>Calc: if [DEBCRE] = &quot;10&quot; then [AMOUNT] else -1* [AMOUNT]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCH</td>
<td>CBGC</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>RECORD</td>
<td>CBGC</td>
<td>N 8</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Calc: precord</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Field Display | Data-file Table Input | Data Length | Entry Status
--- | --- | --- | ---
**Format** Mask
REPLACE CBFC | A 1 | R/W
GROUP CBFC | A 5 | R/W
RGROUP CBFC | A 5 | R/W
**Key:RGROUP**
FILE Len: 5 | Type:A | Dupes:Yes
WORDS Len: 6 | Type:A | Dupes:Yes
PRO_DATE Len: 8 | Type:A | Dupes:Yes
AMOUNT Len: 8 | Type:A | Dupes:Yes
FILE CBFC | A 3 | R/W
WORDS CBFC | A 6 | R/W
DESC1 CBFC | A 4 | R/W
DESC1 CBFC | A 26 | R/W
AMOUNT CBFC | N 8 | R/W
**Key:AMOUNT**
RGROUP Len: 5 | Type:A | Dupes:Yes
DEBCRE Len: 2 | Type:A | Dupes:Yes
DESC1 Len: 4 | Type:A | Dupes:Yes
PRO_DATE Len: 8 | Type:A | Dupes:Yes
DESC2 CBFC | A 12 | R/W
PRO_DATE CBFC | D 8 | R/W
DEBCRE CBFC | A 2 | R/W
**Key:DEBCRE**
DESC1 Len: 4 | Type:A | Dupes:Yes
AMOUNT Len: 8 | Type:A | Dupes:Yes
PRO_DATE Len: 8 | Type:A | Dupes:Yes
SIGN CBFC | N 8 | R/W
**calc:** if [DEBCRE] = "60" then [AMOUNT]
else -1* [AMOUNT]
MATCH CBFC | A 5 | R/W
RECORD CBFC | N 8 | R/W
Calc: precord
View: \u:\alm\u18877\auto\talleyg.vws
Date Created: 11/04/91
Last Edited: 11/21/91
Field Count: 26
Record Count: 1
Encrypted: No

Data-file: \u:\alm\u18877\auto\TALLEYG.db
Total Fields: 26
Record Length: 209
Last Add Date: 11/21/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table</th>
<th>Data Length Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Input Mask</td>
<td>Type Length Entry</td>
</tr>
<tr>
<td>NETGL</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETCB</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETACH</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETCC</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETCL</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETCPN</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETFS</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETFTS</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETINV</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETSB</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETOTHER</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETALL</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>NETUNID</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>GROUP</td>
<td>TALLEYG</td>
<td>A 5 R/W</td>
</tr>
<tr>
<td>CHGCB</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGACH</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGCC</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGCL</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGCPN</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGFS</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGFTS</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGINV</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGOTHER</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGSB</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>CHGALL</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>TALLEYG</td>
<td>N 8 R/W 2R</td>
</tr>
</tbody>
</table>

Calc: [NETGL] - [NETALL]
Field | Data-file Table | Data Length | Entry
---|---|---|---
Display | Input | Type | Status
Format | Mask |  |
NETGL | TALLEYF | N 8 | R/W 2R
NETACH | TALLEYF | N 8 | R/W 2R
NETCC | TALLEYF | N 8 | R/W 2R
NETCL | TALLEYF | N 8 | R/W 2R
NETCPN | TALLEYF | N 8 | R/W 2R
NETFS | TALLEYF | N 8 | R/W 2R
NETFTS | TALLEYF | N 8 | R/W 2R
NETINV | TALLEYF | N 8 | R/W 2R
NETSB | TALLEYF | N 8 | R/W 2R
NETOTHER | TALLEYF | N 8 | R/W 2R
NETALL | TALLEYF | N 8 | R/W 2R
NETUNID | TALLEYF | N 8 | R/W 2R
NETCB | TALLEYF | N 8 | R/W 2R
GROUP | TALLEYF | A 5 | R/W
CHGCB | TALLEYF | N 8 | R/W 2R
CHGACH | TALLEYF | N 8 | R/W 2R
CHGCC | TALLEYF | N 8 | R/W 2R
CHGCCL | TALLEYF | N 8 | R/W 2R
CHGCPN | TALLEYF | N 8 | R/W 2R
CHGFS | TALLEYF | N 8 | R/W 2R
CHGFTS | TALLEYF | N 8 | R/W 2R
CHGINV | TALLEYF | N 8 | R/W 2R
CHGOTHER | TALLEYF | N 8 | R/W 2R
CHGSB | TALLEYF | N 8 | R/W 2R
CHGALL | TALLEYF | N 8 | R/W 2R
DIFFERENCE | TALLEYF | N 8 | R/W 2R

Calc: [NETGL]−[NETALL]
View: u:\alm\u18877\auto\talleyc.vws
Date Created: 12/03/91
Last Edited: 12/02/91
Field Count: 26
Record Count: 1
Encrypted: No

Data-file: u:\alm\u18877\auto\talleyc.db
Total Fields: 26
Record Length: 212
Last Add Date: 12/03/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Mask</td>
<td>Type</td>
<td>Status</td>
</tr>
<tr>
<td>GLCB</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLACH</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLCC</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLCL</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLCPN</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLFS</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLFTS</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLINV</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLOOTHER</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>GLSB</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBCB</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBACH</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBCC</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBCL</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBCPN</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBFS</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBFTS</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBINV</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBOTHER</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>FRBSB</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>TOTALOUT</td>
<td>talleyc</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
</tbody>
</table>

Calc: if ([GLOBAL] < 0 and [FRBBAL] < 0)
then -1*abs((GLOBAL) - [FRBBAL])

else if ([GLOBAL] > 0 and [FRBBAL] > 0)
then abs((GLOBAL) - [FRBBAL])

else
[FRBBAL] - [GLOBAL]
NETDIFF

Calc: if ([NETBAL] < 0 and [TOTALOUT] < 0)
then -1*abs([NETBAL] - [TOTALOUT])

else if ([NETBAL] > 0 and [TOTALOUT] > 0)
then abs([NETBAL] - [TOTALOUT])

else
[NETBAL] + [TOTALOUT]

DATE

talleyc

N  8  R/W  2R

talleyc

D  8  R/W  D1
View: u:\alm\u18877\auto\modg.vws
Date Created: 11/21/91
Last Edited: 11/22/91
Field Count: 16
Record Count: 0
Encrypted: No

Data-file: u:\alm\u18877\auto\modg.db
Total Fields: 16
Record Length: 119
Last Add Date: 12/13/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Mask</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>modg</td>
<td>A</td>
<td>1</td>
<td>R/W</td>
</tr>
<tr>
<td>RCODE</td>
<td>modg</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
</tr>
<tr>
<td>GROUP</td>
<td>modg</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RGROUP</td>
<td>modg</td>
<td>A</td>
<td>7</td>
<td>R/W</td>
</tr>
<tr>
<td>FILE</td>
<td>modg</td>
<td>A</td>
<td>3</td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td>modg</td>
<td>A</td>
<td>6</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1</td>
<td>modg</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1T</td>
<td>modg</td>
<td>A</td>
<td>26</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>modg</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC2</td>
<td>modg</td>
<td>A</td>
<td>12</td>
<td>R/W</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td>modg</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>ENTRY_DATE</td>
<td>modg</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>modg</td>
<td>A</td>
<td>2</td>
<td>R/W</td>
</tr>
<tr>
<td>SIGN</td>
<td>modg</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>MATCH</td>
<td>modg</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RECORD</td>
<td>modg</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Calc: if [WORDS] = "DEBIT" then "10" else "60"

Calc: if [WORDS] = "DEBIT" then [AMOUNT] else -1*[AMOUNT]
**View:** u:\alm\u18877\auto\modfileg.vws
**Date Created:** 11/21/91
**Last Edited:** 11/22/91
**Field Count:** 15
**Record Count:** 0
**Encrypted:** No

**Data-file:** u:\alm\u18877\auto\MODFILEG.db
**Total Fields:** 15
**Record Length:** 112
**Last Add Date:** 11/22/91
**File Format:** Fixed
**Password:** No
**Encrypted:** No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Mask</td>
<td>Type</td>
<td>Status</td>
</tr>
<tr>
<td>REPLACE</td>
<td>MODFILEG</td>
<td>A 1</td>
<td>R/W</td>
</tr>
<tr>
<td>RCODE</td>
<td>MODFILEG</td>
<td>A 7</td>
<td>R/W</td>
</tr>
<tr>
<td>GROUP</td>
<td>MODFILEG</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>Key:GROUP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCODE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGROUP</td>
<td>MODFILEG</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>FILE</td>
<td>MODFILEG</td>
<td>A 3</td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td>MODFILEG</td>
<td>A 6</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1</td>
<td>MODFILEG</td>
<td>A 4</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1T</td>
<td>MODFILEG</td>
<td>A 26</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>MODFILEG</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>DESC2</td>
<td>MODFILEG</td>
<td>A 12</td>
<td>R/W</td>
</tr>
<tr>
<td>EFF_DATE</td>
<td>MODFILEG</td>
<td>D 8</td>
<td>R/W D1</td>
</tr>
<tr>
<td>ENTRY_DATE</td>
<td>MODFILEG</td>
<td>D 8</td>
<td>R/W D1</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>MODFILEG</td>
<td>A 2</td>
<td>R/W</td>
</tr>
<tr>
<td>SIGN</td>
<td>MODFILEG</td>
<td>N 8</td>
<td>R/W 2R</td>
</tr>
<tr>
<td>MATCH</td>
<td>MODFILEG</td>
<td>A 5</td>
<td>R/W</td>
</tr>
</tbody>
</table>
View: u:\alm\u8877\auto\modf.vws
Date Created: 11/21/91
Last Edited: 11/22/91
Field Count: 15
Record Count: 0
Encrypted: No

Data-file: u:\alm\u8877\auto\MODF.db
Total Fields: 15
Record Length: 112
Last Add Date: 12/09/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Mask</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>MODF</td>
<td>A</td>
<td>1</td>
<td>R/W</td>
</tr>
<tr>
<td>RCODE</td>
<td>MODF</td>
<td>A</td>
<td>7</td>
<td>R/W</td>
</tr>
<tr>
<td>GROUP</td>
<td>MODF</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RGROUP</td>
<td>MODF</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>FILE</td>
<td>MODF</td>
<td>A</td>
<td>3</td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td>MODF</td>
<td>A</td>
<td>6</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1</td>
<td>MODF</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1T</td>
<td>MODF</td>
<td>A</td>
<td>26</td>
<td>R/W</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>MODF</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC2</td>
<td>MODF</td>
<td>A</td>
<td>12</td>
<td>R/W</td>
</tr>
<tr>
<td>PRO_DATE</td>
<td>MODF</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>MODF</td>
<td>A</td>
<td>2</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Calc: if [WORDS] = "DEBIT" then "60" else "10"

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGN</td>
<td>MODF</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Calc: if [WORDS] = "DEBIT" then [AMOUNT] else -1*AMOUNT

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCH</td>
<td>MODF</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RECORD</td>
<td>MODF</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
</tbody>
</table>
View: \u:\alm\u18877\auto\modfilef.vws
Date Created: 11/21/91
Last Edited: 11/22/91
Field Count: 14
Record Count: 0
Encrypted: No

Data-file: \u:\alm\u18877\auto\MODFILEF.db
Total Fields: 14
Record Length: 104
Last Add Date: 11/22/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field</th>
<th>Data-file Table</th>
<th>Data</th>
<th>Length</th>
<th>Entry</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td>Mask</td>
<td>Type</td>
<td></td>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>REPLACE</td>
<td>MODFILEF</td>
<td>A</td>
<td>1</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>RCODE</td>
<td>MODFILEF</td>
<td>A</td>
<td>7</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>MODFILEF</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>MODFILEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCODE</td>
<td>MODFILEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>MODFILEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGROUP</td>
<td>MODFILEF</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>MODFILEF</td>
<td>A</td>
<td>3</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>WORDS</td>
<td>MODFILEF</td>
<td>A</td>
<td>6</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>DESC1</td>
<td>MODFILEF</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>DESC1T</td>
<td>MODFILEF</td>
<td>A</td>
<td>26</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>AMOUNT</td>
<td>MODFILEF</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
<td>2R</td>
</tr>
<tr>
<td>DESC2</td>
<td>MODFILEF</td>
<td>A</td>
<td>12</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>PRO_DATE</td>
<td>MODFILEF</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
<td>D1</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>MODFILEF</td>
<td>A</td>
<td>2</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td>MODFILEF</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
<td>2R</td>
</tr>
<tr>
<td>MATCH</td>
<td>MODFILEF</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Data-file Table Input</td>
<td>Data</td>
<td>Length</td>
<td>Entry</td>
<td>Type</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------</td>
<td>------</td>
<td>--------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>GROUP</td>
<td>MATCHES</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>RGROUP</td>
<td>MATCHES</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>MATCHES</td>
<td>A</td>
<td>3</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>WORDS</td>
<td>MATCHES</td>
<td>A</td>
<td>6</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>DESC1</td>
<td>MATCHES</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>DESC1T</td>
<td>MATCHES</td>
<td>A</td>
<td>26</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>DESC2</td>
<td>MATCHES</td>
<td>A</td>
<td>12</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>MATCHES</td>
<td>D</td>
<td>8</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>GLAMOUNT</td>
<td>MATCHES</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
<td>2R</td>
</tr>
<tr>
<td>FRSAMOUNT</td>
<td>MATCHES</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
<td>2R</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>MATCHES</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
<td>2R</td>
</tr>
<tr>
<td>Key: AMOUNT</td>
<td>MATCHES</td>
<td>Len: 8</td>
<td>Type: A</td>
<td>Dupes: Yes</td>
<td></td>
</tr>
<tr>
<td>DESC2</td>
<td>MATCHES</td>
<td>Len: 12</td>
<td>Type: A</td>
<td>Dupes: Yes</td>
<td></td>
</tr>
<tr>
<td>DEBCRE</td>
<td>MATCHES</td>
<td>A</td>
<td>2</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>MATCH</td>
<td>MATCHES</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>RECORD</td>
<td>MATCHES</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>

Calc: precord
### Field Display

<table>
<thead>
<tr>
<th>Format</th>
<th>Mask</th>
<th>Data-file Table Input</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCH</td>
<td>gclear</td>
<td>A 5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>gclear</td>
<td>A 5</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>RGROUP</td>
<td>gclear</td>
<td>A 5</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>gclear</td>
<td>A 3</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>WORDS</td>
<td>gclear</td>
<td>A 6</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>DESC1</td>
<td>gclear</td>
<td>A 4</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>DESCIT</td>
<td>gclear</td>
<td>A 26</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>AMOUNT</td>
<td>gclear</td>
<td>N 8</td>
<td>R</td>
<td></td>
</tr>
</tbody>
</table>

**Key:** AMOUNT

| RGROUP  | Len: 5  | Type: A | Dupes: Yes |
| DESC1   | Len: 4  | Type: A | Dupes: Yes |
| EFF_DATE| Len: 8  | Type: A | Dupes: Yes |

| DESC2   | A 12    | R |
| EFF_DATE| D 8     | R | D1 |
| DEBCRE  | A 2     | R/W |
| D2      | A 4     | R/W |
**View:** \u:\alm\u18877\auto\frbclear.vws

**Date Created:** 12/02/91

**Last Edited:** 12/16/91

**Field Count:** 12

**Record Count:** 0

**Encrypted:** No

**Data-file:** \u:\alm\u18877\auto\frbclear.db

**Total Fields:** 12

**Record Length:** 92

**Last Add Date:** 12/16/91

**File Format:** Fixed

**Password:** No

**Encrypted:** No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Mask</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>MATCH</td>
<td>frbclear</td>
<td>A 5</td>
<td>R/W</td>
</tr>
<tr>
<td>GROUP</td>
<td>frbclear</td>
<td>A 5</td>
<td>R</td>
</tr>
<tr>
<td>RGROUP</td>
<td>frbclear</td>
<td>A 5</td>
<td>R</td>
</tr>
<tr>
<td>FILE</td>
<td>frbclear</td>
<td>A 3</td>
<td>R</td>
</tr>
<tr>
<td>WORDS</td>
<td>frbclear</td>
<td>A 6</td>
<td>R</td>
</tr>
<tr>
<td>DESC1</td>
<td>frbclear</td>
<td>A 4</td>
<td>R</td>
</tr>
<tr>
<td>DESC1T</td>
<td>frbclear</td>
<td>A 26</td>
<td>R</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>frbclear</td>
<td>N 8</td>
<td>R 2R</td>
</tr>
</tbody>
</table>

**Key:** AMOUNT

<table>
<thead>
<tr>
<th>Key: AMOUNT</th>
<th>Mask</th>
<th>Length</th>
<th>Type</th>
<th>Dupes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGROUP</td>
<td></td>
<td>Len: 5</td>
<td>Type:A</td>
<td>Dupes:Yes</td>
</tr>
<tr>
<td>DESC1</td>
<td></td>
<td>Len: 4</td>
<td>Type:A</td>
<td>Dupes:Yes</td>
</tr>
<tr>
<td>PRO_DATE</td>
<td></td>
<td>Len: 26</td>
<td>Type:A</td>
<td>Dupes:Yes</td>
</tr>
</tbody>
</table>

**DESC2**

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRO_DATE</td>
<td>frbclear</td>
<td>D 8</td>
<td>R  D1</td>
</tr>
<tr>
<td>DEBCRE</td>
<td>frbclear</td>
<td>A 2</td>
<td>R/W</td>
</tr>
<tr>
<td>D2</td>
<td>frbclear</td>
<td>A 4</td>
<td>R/W</td>
</tr>
</tbody>
</table>
View: u:\alm\u18877\auto\clear.vws
Date Created: 11/27/91
Last Edited: 11/27/91
Field Count: 14
Record Count: 0
Encrypted: No

Data-file: u:\alm\u18877\auto\CLEAR.db
Total Fields: 14
Record Length: 112
Last Add Date: 11/27/91
File Format: Fixed
Password: No
Encrypted: No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table Input</th>
<th>Data Type</th>
<th>Length</th>
<th>Entry Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format GROUP</td>
<td>CLEAR</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>FILE</td>
<td>CLEAR</td>
<td>A</td>
<td>3</td>
<td>R/W</td>
</tr>
<tr>
<td>WORDS</td>
<td>CLEAR</td>
<td>A</td>
<td>6</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1</td>
<td>CLEAR</td>
<td>A</td>
<td>4</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC1T</td>
<td>CLEAR</td>
<td>A</td>
<td>26</td>
<td>R/W</td>
</tr>
<tr>
<td>DESC2</td>
<td>CLEAR</td>
<td>A</td>
<td>12</td>
<td>R/W</td>
</tr>
<tr>
<td>DEBCRS</td>
<td>CLEAR</td>
<td>A</td>
<td>2</td>
<td>R/W</td>
</tr>
<tr>
<td>MATCH</td>
<td>CLEAR</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RGROUP</td>
<td>CLEAR</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>RECORD</td>
<td>CLEAR</td>
<td>N</td>
<td>8</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Calc: precord

DATE          | CLEAR                 | D         | 8      | R/W          | D1  |
GLAMOUNT      | CLEAR                 | N         | 8      | R/W          | 2R  |
FRBAMOUNT     | CLEAR                 | N         | 8      | R/W          | 2R  |
AMOUNT        | CLEAR                 | N         | 8      | R/W          | 2R  |

Key: AMOUNT
DESC2

Len: 8 Type:A Dupes:Yes
Len: 12 Type:A Dupes:Yes
**View:** u:\alm\u18877\auto\merge.vws  
**Date Created:** 11/22/91  
**Last Edited:** 12/13/91  
**Field Count:** 12  
**Record Count:** 0  
**Encrypted:** No  

**Data-file:** u:\alm\u18877\auto\MERGE.db  
**Total Fields:** 12  
**Record Length:** 96  
**Last Add Date:** 12/13/91  
**File Format:** Fixed  
**Password:** No  
**Encrypted:** No

<table>
<thead>
<tr>
<th>Field Display</th>
<th>Data-file Table</th>
<th>Data Length</th>
<th>Entry Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format GROUP</td>
<td>Mask MERGE</td>
<td>A</td>
<td>5</td>
<td>R/W</td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 5</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 5</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 5</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 5</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 3</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:GROUP</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 3</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 3</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 3</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 3</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 6</td>
<td>Type:D</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
<tr>
<td>Key:DATE</td>
<td>Len: 8</td>
<td>Type:A</td>
<td>Dupe:Yes</td>
<td></td>
</tr>
</tbody>
</table>

**Calc:** if [DEBCRE] = "10" then [AMOUNT]  
else -1*[AMOUNT]  

**MATCH**  

**SIGN**  

**AMOUNT**  

**DESC2**  

**DESC1**  

**DESC1T**  

**DEBCRE**  

**DATE**  

**WORDS**  

**FILE**  

**RGROUP**  

**GROUP**  

**VIEW**
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

TEST SCRIPT

Test Script for Cycle 1: Implementation of FIE / Confirmation of Transactions

Account Reconciliation Project

Dominion Bankshares Corporation
OK

1. Verify account assignment to group.
2. Check the individual G/L items to see if the correct group is assigned.
3. Verify newly assigned transactions are in the correct group.
4. Assign the additional FRP files to see if any new transactions are assigned to the additional group.
5. Test the additional group to verify the transactions are properly grouped.

OK

6. Check the merged G/L items to see if all transactions are properly grouped.
7. Test the merged group to verify all transactions are properly grouped.
8. Assign an individual FRP file to see if the transactions are properly grouped.
9. Assign an individual table to see if the transactions are properly grouped.
10. Any transaction that is not in the correct group needs to be reassigned.

OK

11. Move to modification cycle.
12. Sign-off Comments:

Data

Sign-off

Step/Condition Tested

Test Script for Cycle 2: Assigning Group Codes
Account Reconciliation Process
Dominion Bankshares Corporation
OK

1. Remove all items marked "not physically examined" from the list.
2. Verify the transactions listed in the report.
3. Execute the report to list all matched items.
4. Ensure all transactions are listed.
5. Sign the report.

Test steps:
- Step 1: Verify the transactions listed in the report.
- Step 2: Execute the report to list all matched items.
- Step 3: Ensure all transactions are listed.
- Step 4: Sign the report.

Dominion Bankshares Corporation
Account Recommission Project

Comments

Sign-Off

Data

#
Test Set Up for Cycle 3: Cleaning Transactions

Dominion Bankshares Corporation
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Print the Cover Sheet Report Check the Functional Use of the Item.</td>
</tr>
<tr>
<td>6.2</td>
<td>Print the Detailed Items Report. Evaluate the Functional Use of the Item.</td>
</tr>
<tr>
<td>6.3</td>
<td>Check the Report (for Formulating the Functional Use).</td>
</tr>
<tr>
<td>6.5</td>
<td>Evaluate the Detailed Items Report.</td>
</tr>
<tr>
<td>6.6</td>
<td>Print the Report (for Formulating the Functional Use). Check the Functional Use.</td>
</tr>
</tbody>
</table>

Data

**Comments**

Trial Item counts, spelling, examination of headers, dates, formatting checks will include.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Execute the assigned matching <strong>ALL</strong> files.</td>
</tr>
<tr>
<td>7.4</td>
<td>Execute the designated matching <strong>ALL</strong> files.</td>
</tr>
<tr>
<td>7.5</td>
<td>Group code: Assign all unmatched transactions a Test file.</td>
</tr>
<tr>
<td>7.6</td>
<td>Import two new G/L and See file. Test files.</td>
</tr>
<tr>
<td>7.7</td>
<td>Step # of Condition Tested</td>
</tr>
</tbody>
</table>
PROGRAM DESCRIPTION
DBC GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM NAME: GL ACCOUNT 1060 AUTOMATED MATCHING PROGRAM

PROGRAM LISTING
global $choice
main
file unload all
label menu
repaint on
screen clear box 1 1 21 80 0 5
screen clear box 2 14 20 67 0 0
screen draw box 2 14 20 67 15 0
screen print 4 25 11 0 "GL Account 1060 Reconciliation"
screen print 5 27 11 0 "Automated Matching Program"
screen print 6 27 11 0 "++++++++++++++++++++++"
screen print 9 25 3 0 "A) Import Files"
screen print 10 25 3 0 "B) Modify Group Files"
screen print 11 25 3 0 "C) Automated Matching by Group"
screen print 12 25 3 0 "D) Clearing by Group"
screen print 13 25 3 0 "E) Reports Menu"
screen print 14 25 3 0 "F) Restore Prior Day's Files"
screen print 15 25 3 0 "G) Back Up Files to Diskette"
screen print 16 25 3 0 "X) Exit"
screen print 18 25 3 0 "Choice: "
screen input 18 33 3 0 1 $choice mask "[A-?x-a-?x]"
case upper($choice)
  when "A"
    execute "import"
    jump menu
  when "B"
    transfer "menu mod"
    jump menu
  when "C"
    transfer "menu match"
    jump menu
  when "D"
    execute "menu clr"
    jump menu
  when "E"
    transfer "rpt menu"
    when "F"
      execute "restore"
      jump menu
    when "G"
      execute "backup"
      jump menu
  otherwise
    stop
end case
end main
IMPORT.PF3

'Files that must be downloaded from the mainframe:
'UCS_SEC.FED.RECON.DOWNLOAD - file name is FEDFILE
'UCS.GLJ.GWP12.DOWNLOAD.ACCT1061 - file name is GLFILE

'Variable declaration
global $netcb $netach $netcc $netcl $netfps $netfts $netinv $netsb
global $netother $netgl $netall $netunid
main
quiet off

'Clear screen
repaaint off
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 30 3 0 "Importing GL, please wait . . . ."

'Copy all group file to subdirectory C:\restore
tools file copy "c:\auto\cbgc.*" to "c:\auto\restore"
tools file copy "c:\auto\cbfc.*" to "c:\auto\restore"
tools file copy "c:\auto\achqc.*" to "c:\auto\restore"
tools file copy "c:\auto\achfc.*" to "c:\auto\restore"
tools file copy "c:\auto\ccgc.*" to "c:\auto\restore"
tools file copy "c:\auto\ccfc.*" to "c:\auto\restore"
tools file copy "c:\auto\clgc.*" to "c:\auto\restore"
tools file copy "c:\auto\clfc.*" to "c:\auto\restore"
tools file copy "c:\auto\cpngc.*" to "c:\auto\restore"
tools file copy "c:\auto\cpnfc.*" to "c:\auto\restore"
tools file copy "c:\auto\fsgc.*" to "c:\auto\restore"
tools file copy "c:\auto\ssfc.*" to "c:\auto\restore"
tools file copy "c:\auto\ftsgc.*" to "c:\auto\restore"
tools file copy "c:\auto\ftsf.*" to "c:\auto\restore"
tools file copy "c:\auto\invgc.*" to "c:\auto\restore"
tools file copy "c:\auto\invfc.*" to "c:\auto\restore"
tools file copy "c:\auto\othergc.*" to "c:\auto\restore"
tools file copy "c:\auto\otherfc.*" to "c:\auto\restore"
tools file copy "c:\auto\sbgc.*" to "c:\auto\restore"
tools file copy "c:\auto\sbfc.*" to "c:\auto\restore"
tools file copy "c:\auto\talleyg.*" to "c:\auto\restore"
tools file copy "c:\auto\talleyf.*" to "c:\auto\restore"
'GL file import and subdivide into group files
' Erase group files created during previous execution

tools file erase "CBG.DB"
tools file erase "CBG.VW"
tools file erase "CBG.KEY"

tools file erase "ACHG.DB"
tools file erase "ACHG.VW"
tools file erase "ACHG.KEY"

tools file erase "CCG.DB"
tools file erase "CCG.VW"
tools file erase "CCG.KEY"

tools file erase "CLG.DB"
tools file erase "CLG.VW"
tools file erase "CLG.KEY"

tools file erase "CPNG.DB"
tools file erase "CPNG.VW"
tools file erase "CPNG.KEY"

tools file erase "FSG.DB"
tools file erase "FSG.VW"
tools file erase "FSG.KEY"

tools file erase "FTSG.DB"
tools file erase "FTSG.VW"
tools file erase "FTSG.KEY"

tools file erase "INVG.DB"
tools file erase "INVG.VW"
tools file erase "INVG.KEY"

tools file erase "SBG.DB"
tools file erase "SBG.VW"
tools file erase "SBG.KEY"

tools file erase "OTHERG.DB"
tools file erase "OTHERG.VW"
tools file erase "OTHERG.KEY"

tools file erase "UNIDG.DB"
tools file erase "UNIDG.VW"
tools file erase "UNIDG.KEY"

' Delete existing records from GL file
 file load standard-view "gl.vws"
data query execute "delall.dfq"
' replace delete
keys Esc
file unload all
data utilities purge "gl.db"

' Import records into GL file
file load standard-view "gl.vws"
file import ascii "a:\glfile" fields
"[COMPANY;ACCOUNT;CENTER;SOURCE;DESC_1;DESC_2;AMOUNT;EFF_DATE;ENTRY_DATE;
DEBCRE]"

data utilities recalcalc-all

'delete records from account #1036
data query execute "1036del.dfq"
'[ACCOUNT]="1036" REPLACE DELETE
keys Esc

'delete records with zero amount
data query execute "zero.dfq"
'[SIGN]=0 REPLACE DELETE
keys Esc

file unload all
data utilities purge "gl.db"
file load standard-view "gl.vws"
$netgl = filesum([sign])
window split vertical 34
data goto window 2

'create CB group file
file create "CBG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"CBG",Enter,Enter,Enter
keys F10
data goto window 1
data query execute "CB" index "CB"
'[GROUP]="CB"
keys Enter,"CB",Enter,Esc
'check for existence of CB records
if cerror <> 3143
  data goto window 2
data utilities append "gl.vws"
keys F10
  $netcb = filesum([sign])
else
  data goto window 2
  $netcb = 0
end if
file unload view "CBG.vw"

'create ACH group file
file create "ACHG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"ACHG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "ACH" index "ACH"
'[GROUP]="ACH"
keys Enter,"ACH",Enter,Esc
'check for existence of ACH records
if cerror <> 3141
  data goto window 2
data utilities append "gl.vws"
keys F10
  $netach = filesum([sign])
else

data goto window 2
$netach = 0
end if
file unload view "ACHG.vw"

'Create CC group file
file create "CCG" similar standard-view "g1group.vws" no-password
keys "RD", Enter, "CCG", Enter, Enter, Enter
keys F10
data goto window 1
order change physical
data query execute "cc.dfq" index "CC"
'GROUP"="CC"
keys Enter, "CC", Enter, Esc
'Check for existence of CC records
if cerror <> 3143
data goto window 2
data utilities append "gl.vws"
keys F10
$netcc = filesum([sign])
else
data goto window 2
$netcc = 0
end if
file unload view "CCG.vw"

'Create CL group file
file create "CLG" similar standard-view "glgroup.vws" no-password
keys "RD", Enter, "CLG", Enter, Enter, Enter
keys F10
data goto window 1
order change physical
data query execute "CL" index "CL"
'GROUP"="CL"
keys Enter, "CL", Enter, Esc
'Check for existence of CL records
if cerror <> 3143
data goto window 2
data utilities append "gl.vws"
keys F10
$netcl = filesum([sign])
else
data goto window 2
$netcl = 0
end if
file unload view "CLG.vw"

'Create CPN group file
file create "CPNG" similar standard-view "g1group.vws" no-password
keys "RD", Enter, "CPNG", Enter, Enter, Enter
keys F10
data goto window 1
order change physical
data query execute "CPN" index "CPN"
'GROUP"="CPN"
keys Enter, "CPN", Enter, Esc
'Check for existence of CPN group records
if cerror <> 3143
data goto window 2
data utilities append "gl.vws"
keys F10
$netcpn = filesum([sign])
else
data goto window 2
$netcpn = 0
end if
file unload view "CPNG.vw"

'Create FS group file
file create "FSG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"FSG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "FS" index "FS"
'[GROUP]="FS"
keys Enter,"FS",Enter,Esc
'Check for existence of FS records
if error <> 3143
data goto window 2
data utilities append "gl.vws"
keys F10
$netfs = filesum([sign])
else
data goto window 2
$netfs = 0
end if
file unload view "FSG.vw"

'Create FTS group file
file create "FTSG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"FTSG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "FTS" index "FTS"
'[GROUP]="FTS"
keys Enter,"FTS",Enter,Esc
'Check for existence of FTS records
if error <> 3143
data goto window 2
data utilities append "gl.vws"
keys F10
$netfts = filesum([sign])
else
data goto window 2
$netfts = 0
end if
file unload view "FTSG.vw"

'Create INV group file
file create "INVG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"INVG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "INV" index "INV"
' [GROUP]="INV"
keys Enter,"INV",Enter,Esc
' Check for existence of INV records
if cerror <> 3143
  data goto window 2
  data utilities append "gl.vws"
  keys F10
  $netinv = filesum([sign])
else
  data goto window 2
  $netinv = 0
end if
file unload view "INVG.vw"

' Create SB group file
file create "SBG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"SBG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "SB" index "SB"
' [GROUP]="SB"
keys Enter,"SB",Enter,Esc
' Check for existence of SB records
if cerror <> 3143
  data goto window 2
  data utilities append "gl.vws"
  keys F10
  $netsb = filesum([sign])
else
  data goto window 2
  $netsb = 0
end if
file unload view "SBG.vw"

' Create OTHER group file
file create "OTHERG" similar standard-view "glgroup.vws" no-password
keys "RD",Enter,"OTHERG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "OTHER" index "OTHER"
' [GROUP]="OTHER"
keys Enter,"OTHER",Enter,Esc
' Check for existence of OTHER records
if cerror <> 3143
  data goto window 2
  data utilities append "gl.vws"
  keys F10
  $netother = filesum([sign])
else
  data goto window 2
  $netother = 0
end if
file unload view "OTHERG.vw"

' Create unidentified items group file
file create "UNIDG" similar standard-view "achgc.vws" no-password
keys "RD",Enter,"UNIDG",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "UNID" index "UNID"
'[GROUP]="?????"
keys Enter,"UNID",Enter,Esc
'Check for existence of unidentified records
if cerror <> 3143
data goto window 2
data utilities append "gl.vws"
keys F10
$netunid = filesum([sign])
data query execute "empty_g.dfq"
keys Esc
else
data goto window 2
$netunid = 0
end if
file unload all
window close

'Assign totals to fields in TALLEY file
file load standard-view "talleyg.vws"
data goto record first
[NETGL] = $netgl
[NETCB] = $netcb
[NETACH] = $netach
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netfs
[NETFTS] = $netfts
[NETINV] = $netinv
[NETSB] = $netsb
[NETOTHER] = $netother
[NETUNID] = $netunid
$nega = 1
$netcb+$netach+$netcc+$netcpn+$netfs+$netfts+$netinv+$netsb+$netother+$netunid
[NETALL] = $netall

'Verify that net GL activity is equal to net group activity
if round($netgl,2) = round($netall,2)
    print report execute "groupgog.dfr" printer detail start 1 end 0 copies 1
    file load standard-view "unidg.vw"
    order sort execute "debcres.dfs" index "debcres.idx"
    keys NoKey,Esc
    if cerror <> 3958
        print report execute "ug.dfr" printer detail start 1 end 0 copies 1
        file unload all
        execute "assigng"
    else
        jump label1
    end if
else

print report execute "grpnogog.dfr" printer detail start 1 end 0
copies 1
end if

label labell
file unload all

'FRB file import and subdivide into group files
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 30 3 0 "Importing FRB, please wait . . . ."

'Erase group files created during previous execution
tools file erase "CBF.DB"
tools file erase "CBF.VW"
tools file erase "CBF.KEY"
tools file erase "ACHF.DB"
tools file erase "ACHF.VW"
tools file erase "ACHF.KEY"
tools file erase "CCF.DB"
tools file erase "CCF.VW"
tools file erase "CCF.KEY"
tools file erase "CLF.DB"
tools file erase "CLF.VW"
tools file erase "CLF.KEY"
tools file erase "CPNF.DB"
tools file erase "CPNF.VW"
tools file erase "CPNF.KEY"
tools file erase "FSF.DB"
tools file erase "FSF.VW"
tools file erase "FSF.KEY"
tools file erase "FTSF.DB"
tools file erase "FTSF.VW"
tools file erase "FTSF.KEY"
tools file erase "INVF.DB"
tools file erase "INVF.VW"
tools file erase "INVF.KEY"
tools file erase "SBF.DB"
tools file erase "SBF.VW"
tools file erase "SBF.KEY"
tools file erase "OTHERF.DB"
tools file erase "OTHERF.VW"
tools file erase "OTHERF.KEY"
tools file erase "UNIDF.DB"
tools file erase "UNIDF.VW"
tools file erase "UNIDF.KEY"

'Delete existing records from FRB file
file load standard-view "frb.vws"
data query execute "delall.dfq"
'replace delete
keys Esc
file unload all
data utilities purge "frb.db"

'Import records into FRB file
file load standard-view "frb.vws"
file import fixed " A:\FEDFILE" fields
"[ABA;STMT_SEC;TRANS_CODE;DC_CODE;AMOUNT;REF;PRO_DATE;AVA_DATE;CON_ITEM;
BANK_DFI;DEBCRE;DESCI;DESC17]"
data utilities recalc-all

'Delete deferred records
data query execute "deferred.dfq"
'[STMT_SEC]="D" REPLACE DELETE
keys Esc

'Delete records with zero amount
data query execute "zero.dfq"
'[SIGN]=0 REPLACE DELETE
keys Esc
file unload view "frb.vws"
data utilities purge "frb.db"
file load standard-view "frb.vws"

$netg1 = filesum([sign])
window split vertical 34
data goto window 2

'Create CB group file
file create "CBF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"CBF",Enter,Enter
keys F10
data goto window 1
data query execute "CB" index "CB"
'[GROUP]="CB"
keys Enter,"CB",Enter,Esc
'Check for existence of CB records
if errror <> 3143
data goto window 2
data utilities append "frb.vws"
keys F10
$netcb = filesum([sign])
else
data goto window 2
$netcb = 0
end if
file unload view "CBF.vw"

'Create ACH group file
file create "ACHF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"ACHF",Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "ACH" index "ACH"
 GIR"="ACH"
keys Enter,"ACH",Enter,Esc
'Check for existence of ACH records
if cerror <> 3143
  data goto window 2
data utilities append "frb.vws"
  keys F10
  $netach = filesum([sign])
else
  data goto window 2
  $netach = 0
end if
file unload view "ACHF.vw"

'Create CC group file
file create "CCF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"CCF",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "cc.dfg" index "CC"
'GIR"="CC"
keys Enter,"CC",Enter,Esc
'Check for existence of CC records
if cerror <> 3143
  data goto window 2
data utilities append "frb.vws"
  keys F10
  $netcc = filesum([sign])
else
  data goto window 2
  $netcc = 0
end if
file unload view "CCF.vw"

'Create CL group file
file create "CLF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"CLF",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "CL" index "CL"
'GIR"="CL"
keys Enter,"CL",Enter,Esc
'Check for existence of CL records
if cerror <> 3143
  data goto window 2
data utilities append "frb.vws"
  keys F10
  $netcl = filesum([sign])
else
  data goto window 2
  $netcl = 0
end if
file unload view "CLF.vw"
'Create CPN group file
file create "CPNF" similar standard-view "frbgroupl.vws" no-password
keys "RD", Enter, "CPNF", Enter, Enter, Enter
keys F10
data goto window 1
order change physical
data query execute "CPN" index "CPN"
'[GROUP]="CPN"
keys Enter, "CPN", Enter, Esc
'Check for existence of CPN group records
if cerror <> 3143
data goto window 2
data utilities append "frb.vws"
keys F10
$netcpn = filesum([sign])
elser
data goto window 2
$netcpn = 0
end if
file unload view "CPNF.vw"

'Create FS group file
file create "FSF" similar standard-view "frbgroupl.vws" no-password
keys "RD", Enter, "FSF", Enter, Enter, Enter
keys F10
data goto window 1
order change physical
data query execute "FS" index "FS"
'[GROUP]="FS"
keys Enter, "FS", Enter, Esc
'Check for existence of FS records
if cerror <> 3143
data goto window 2
data utilities append "frb.vws"
keys F10
$netfs = filesum([sign])
elser
data goto window 2
$netfs = 0
end if
file unload view "FSF.vw"

'Create FTS group file
file create "FTSF" similar standard-view "frbgroupl.vws" no-password
keys "RD", Enter, "FTSF", Enter, Enter, Enter
keys F10
data goto window 1
order change physical
data query execute "FTS" index "FTS"
'[GROUP]="FTS"
keys Enter, "FTS", Enter, Esc
'Check for existence of FTS records
if cerror <> 3143
data goto window 2
data utilities append "frb.vws"
keys F10
$netfts = filesum([sign], [GROUP]="FTS")
elser

data goto window 2
$netfts = 0
end if
file unload view "FTSF.vw"

'Create INV group file
file create "INVF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"INVF",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "INV" index "INV"
'[GROUP]="INV"
keys Enter,"INV",Enter,Esc
'Check for existence of INV records
if cerror <> 3143
  data goto window 2
data utilities append "frb.vws"
  keys F10
  $netinv = filesum([sign])
else
  data goto window 2
  $netinv = 0
end if
file unload view "INVF.vw"

'Create SB group file
file create "SBF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"SBF",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "SB" index "SB"
'[GROUP]="SB"
keys Enter,"SB",Enter,Esc
'Check for existence of SB records
if cerror <> 3143
  data goto window 2
data utilities append "frb.vws"
  keys F10
  $netsb = filesum([sign])
else
  $netsb = 0
  data goto window 2
end if
file unload view "SBF.vw"

'Create OTHER group file
file create "OTHERF" similar standard-view "frbgroup.vws" no-password
keys "RD",Enter,"OTHERF",Enter,Enter,Enter
keys F10
data goto window 1
order change physical
data query execute "OTHER" index "OTHER"
'[GROUP]="OTHER"
keys Enter,"OTHER",Enter,Esc
'Check for existence of OTHER records
if cerror <> 3143
data goto window 2
data utilities append "frb.vws"
keys F10
$netother = filesum([sign])
else
data goto window 2
$netother = 0
end if
file unload view "OTHERF.vw"

'Create unidentified items group file
file create "UNIDF" similar standard-view "achfc.vws" no-password
keys "RD",Enter,"UNIDF",Enter,Enter,Esc
keys F10
data goto window 1
order change physical
data query execute "UNID" index "UNID"
'[GROUP]="??????"
keys Enter,"UNID",Enter,Esc
'Check for existence of unidentified records
if cerror <> 3143
data goto window 2
data utilities append "frb.vws"
keys F10
$netunid = filesum([sign])
data query execute "empty_g.dfq"
keys Esc
else
data goto window 2
$netunid = 0
end if
file unload all
window close

'Assign totals to fields in TALLEY file
file load standard-view "talleyf.vws"
data goto record first
[NETG1] = $netg1
[NETC1] = $netc1
[NETACH] = $netach
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netfs
[NETFTS] = $netfts
[NETINV] = $netinv
[NETSB] = $netsb
[NETOTHER] = $netother
[NETUNID] = $netunid
$netc1+nettach+netcc+netcl+netcpn+netfs+netfts+netinv+netsb+netother+netunid
[NETALL] = $netall

'Verify that net FRB activity is equal to net group activity
if round($netg1,2) = round($netall,2)
    print report execute "groupof.dfr" printer detail start 1 end 0
copies 1
file load standard-view "unidf.vw"
order sort execute "debcrc.dfs" index "debcrc.idx"
keys NoKey,Esc
if cerror <> 3958
print report execute "uf.dfr" printer detail start 1 end 0 copies 1
file unload all
execute "assignf"
else
screen clear 3 0
screen draw box 8 18 12 59 3 0
screen print 10 20 5 0 "No unidentified items from FRB file . .
wait 2
screen print 10 20 5 0 "Merging FRB current & cumulative files

window split vertical 42
file load standard-view "cbf.vw"
data goto window 2
file load standard-view "cbfc.vws"
data utilities append "cbf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netch = filesize([sign])
data goto window 1
file unload all
else
$netch = filesize([sign])
data goto window 1
file unload all
end if

file load standard-view "achf.vw"
data goto window 2
file load standard-view "achfc.vws"
data utilities append "achf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netch = filesize([sign])
data goto window 1
file unload all
else
$netch = filesize([sign])
data goto window 1
file unload all
end if

file load standard-view "ccf.vw"
data goto window 2
file load standard-view "ccfc.vws"
data utilities append "ccf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netcc = filesize([sign])
data goto window 1
file unload all
else
\$netcc = filesum([sign])
  data goto window 1
  file unload all
end if

file load standard-view "clf.vw"
data goto window 2
file load standard-view "clfc.vws"
data utilities append "clf.vw"
keys F10, NoKey, Esc
if cerror = 3958
  \$netcl = filesum([sign])
data goto window 1
  file unload all
else
  \$netcl = filesum([sign])
data goto window 1
  file unload all
end if

file load standard-view "cpnf.vw"
data goto window 2
file load standard-view "cpnfc.vws"
data utilities append "cpnf.vw"
keys F10, NoKey, Esc
if cerror = 3958
  \$netcpn = filesum([sign])
data goto window 1
  file unload all
else
  \$netcpn = filesum([sign])
data goto window 1
  file unload all
end if

file load standard-view "fsf.vw"
data goto window 2
file load standard-view "fsfc.vws"
data utilities append "fsf.vw"
keys F10, NoKey, Esc
if cerror = 3958
  \$netfs = filesum([sign])
data goto window 1
  file unload all
else
  \$netfs = filesum([sign])
data goto window 1
  file unload all
end if

file load standard-view "ftsfc.vws"
data utilities append "ftsf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netfts = filesum([sign])
data goto window 1
file unload all
else
$netfts = filesum([sign])
data goto window 1
file unload all
end if

file load standard-view "invf.vw"
data goto window 2
file load standard-view "invfc.vws"
data utilities append "invf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netinv = filesum([sign])
data goto window 1
file unload all
else
$netinv = filesum([sign])
data goto window 1
file unload all
end if

file load standard-view "sbf.vw"
data goto window 2
file load standard-view "sbfc.vws"
data utilities append "sbf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netsb = filesum([sign])
data goto window 1
file unload all
else
$netsb = filesum([sign])
data goto window 1
file unload all
end if

file load standard-view "otherf.vw"
data goto window 2
file load standard-view "otherfc.vws"
data utilities append "otherf.vw"
keys F10,NoKey,Esc
if cerror = 3958
$netother = filesum([sign])
data goto window 1
file unload all
else
$netother = filesum([sign])
data goto window 1
file unload all
end if
window close

file load standard-view "talleyf.vws"
data goto record first
$ete$ = $netcb+$netach+$netcc+$netcl+$netcpn+$netfs+$netfts+$netinv+$netsb+$netother+$netunid
[NETGL] = $netall
[NETALL] = $netall
[NETCB] = $netcb
[NETACH] = $netach
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netfs
[NETFTS] = $netfts
[NETINV] = $netinv
[NETSB] = $netsb
[NETOTHER] = $netother
[NETUNID] = 0
file unload all

tools file erase "CBF.DBF"
tools file erase "CBF.VW"
tools file erase "CBF.KEY"
tools file erase "ACHF.DBF"
tools file erase "ACHF.VW"
tools file erase "ACHF.KEY"
tools file erase "CCF.DBF"
tools file erase "CCF.VW"
tools file erase "CCF.KEY"
tools file erase "CLF.DBF"
tools file erase "CLF.VW"
tools file erase "CLF.KEY"
tools file erase "CPNF.DBF"
tools file erase "CPNF.VW"
tools file erase "CPNF.KEY"
tools file erase "FSF.DBF"
tools file erase "FSF.VW"
tools file erase "FSF.KEY"
tools file erase "FTSF.DBF"
tools file erase "FTSF.VW"
tools file erase "FTSF.KEY"
tools file erase "INVF.DBF"
tools file erase "INVF.VW"
tools file erase "INVF.KEY"
tools file erase "SBF.DBF"
tools file erase "SBF.VW"
tools file erase "SBF.KEY"

tools file erase "OTHERF.DB"
tools file erase "OTHERF.VW"
tools file erase "OTHERF.KEY"

tools file erase "UNIDF.DB"
tools file erase "UNIDF.VW"
tools file erase "UNIDF.KEY"

jump label2
    end if

else
    print report execute "grpnogof.dfr" printer detail start 1 end 0
copies 1
    end if

label label2
file unload all
end main
ASSIGN0.PP3

'Variable definition
global $addcb $addach $addcc $addcl $addcpp $adddfs
global $addfts $addinv $addsb $addother
global $newcb $newach $newcc $newcl $newcpp $newdfs
global $newfts $newinv $newsb $newother
global $netgl $newgl
global $cumcb $cuxcb $cumach $cuxach $cumcc $cuxcc $cumcl $cuxcl
global $cumcpp $cuxcpp $cumfs $cuxfs $cumfts $cuxfts $cuminv $cuxinv
global $cumsb $cuxsb $cumother $cuxother
main

repaint off
screen clear 3 0
screen print 10 20 5 0 "Please wait . . ."
screen draw box 8 18 12 59 3 0

'Check if unidentified items file exists, if it does not then this routine
'has already been run for today and should not be run again.
    if file("c:\auto\unidg.db") = 0
        screen clear 3 0
        screen print 10 20 3 0 "Unidentified GL items have already been
assigned"
        wait 2.5
        jump out
    else

label edit

'Load unidentified items file, bring onto screen for group code assignment
    file load standard-view "unidg.vw"
    order sort execute "debcf.dfs" index "debcf.idx"
repaint on

data enter
repaint off
screen clear 3 0
screen print 10 20 5 0 "Please wait . . ."
screen draw box 8 18 12 59 3 0

'Unassigned items are given code OTHER

data query execute "nogroup.dfg"
'([GROUP]<"CB" AND [GROUP]<"ACH" AND [GROUP]<"CC" AND [GROUP]<"CL"
'AND [GROUP]<"CPN" AND [GROUP]<"FS" AND [GROUP]<"FTS" AND
'[GROUP]<"INV" AND [GROUP]<"SB" AND [GROUP]<"OTHER")
'REPLACE [GROUP]="OTHER"
keys Esc

'Calculate net activity for each group

data query execute "CB" index "CB"
keys Esc
if cerror <> 3143
$addcb = filesize([sign])
else
    $addcb = 0
end if
order change physical

data query execute "ACH" index "ACH"
keys Esc
if cerror <> 3143
    $addach = filesize([sign])
else
    $addach = 0
end if
order change physical

data query execute "CC" index "CC"
keys Esc
if cerror <> 3143
    $addcc = filesize([sign])
else
    $addcc = 0
end if
order change physical

data query execute "CL" index "CL"
keys Esc
if cerror <> 3143
    $addcl = filesize([sign])
else
    $addcl = 0
end if
order change physical

data query execute "CPN" index "CPN"
keys Esc
if cerror <> 3143
    $addcpn = filesize([sign])
else
    $addcpn = 0
end if
order change physical

data query execute "FS" index "FS"
keys Esc
if cerror <> 3143
    $addfs = filesize([sign])
else
    $addfs = 0
end if
order change physical

data query execute "FTS" index "FTS"
keys Esc
if cerror <> 3143
  $addfts = $filesum([sign])
else
  $addfts = 0
end if
order change physical

data query execute "INV" index "INV"
keys Esc
if cerror <> 3143
  $addinv = $filesum([sign])
else
  $addinv = 0
end if
order change physical

data query execute "SB" index "SB"
keys Esc
if cerror <> 3143
  $addsb = $filesum([sign])
else
  $addsb = 0
end if
order change physical

data query execute "OTHER" index "OTHER"
keys Esc
if cerror <> 3143
  $addother = $filesum([sign])
else
  $addother = 0
end if
order change physical

'Add new group activity to previous group activity

file load standard-view "TALLEYG.vws"
data goto record first
$netgl = [NETGL]
$newcb = $addcb + [NETCB]
$newach = $addach + [NETACH]
$newcc = $addcc + [NETCC]
$newcl = $addcl + [NETCL]
$newcpn = $addcpn + [NETCPN]
$newfs = $addfs + [NETFS]
$newfts = $addfts + [NETFTS]
$newinv = $addinv + [NETINV]
$newsb = $addsb + [NETSB]
$newother = $addother + [NETOTHER]

$newgl = $newcb + $newach + $newcc + $newcl + $newcpn + $newfs + $newfts + $newinv + $newsb + $newother
'If net totals agree, print report and send records to group files

if round($newgl,2) = round($netgl,2)
data goto record first
[NETCB] = $newcb
[NETACH] = $newach
[NETCC] = $newcc
[NETCL] = $newcl
[NETCPN] = $newcpn
[NETFS] = $newfs
[NETFTS] = $newfts
[NETINV] = $newinv
[NETSB] = $newsb
[NETOTHER] = $newother
[NETUNID] = 0
[NETALL] = $newgl

print report execute "groupgog.dfr" printer detail start 1 end 0 copies 1
data goto view "unidg.vw"
window split vertical 42
data goto window 2

'Send to CB group file
file load standard-view "CBG.vw"
data goto window 1
order change physical
data query execute "CB" index "CB"
'[GROUP]="CB"
keys Enter,"CB",Enter,Esc
'Check for existence of CB records
if cerror <> 3143
data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
file unload view "CBG.vw"

'Send to ACH group file
file load standard-view "ACHG.vw"
data goto window 1
order change physical
data query execute "ACH" index "ACH"
'[GROUP]="ACH"
keys Enter,"ACH",Enter,Esc
'Check for existence of ACH records
if cerror <> 3143
data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
file unload view "ACHG.vw"
'Send to CC group file
file load standard-view "CCG.vw"
data goto window 1
order change physical
data query execute "CC" index "CC"
'\[GROUP]\="CC"
keys Enter,"CC",Enter,Esc
'Check for existence of CC records
if cerror <> 3143
  data goto window 2
  data utilities append "unidg.vw"
  keys F10
else
  data goto window 2
end if
file unload view "CCG.vw"

'Send to CL group file
file load standard-view "CLG.vw"
data goto window 1
order change physical
data query execute "CL" index "CL"
'\[GROUP]\="CL"
keys Enter,"CL",Enter,Esc
'Check for existence of CL records
if cerror <> 3143
  data goto window 2
  data utilities append "unidg.vw"
  keys F10
else
  data goto window 2
end if
file unload view "CLG.vw"

'Send to CPN group file
file load standard-view "CPNG.vw"
data goto window 1
order change physical
data query execute "CPN" index "CPN"
'\[GROUP]\="CPN"
keys Enter,"CPN",Enter,Esc
'Check for existence of CPN group records
if cerror <> 3143
  data goto window 2
  data utilities append "unidg.vw"
  keys F10
else
  data goto window 2
end if
file unload view "CPNG.vw"

'Send to FS group file
file load standard-view "FSG.vw"
data goto window 1
order change physical
data query execute "FS" index "FS"
'\[GROUP]\="FS"
keys Enter,"FS",Enter,Esc
'Check for existence of FS records
if cerror <> 3143
  data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
file unload view "FSG.vw"

'Send to FTS group file
file load standard-view "FTSG.vw"
data goto window 1
order change physical
data query execute "FTS" index "FTS"
'GROUP"="FTS"
keys Enter,"FTS",Enter,Esc
'Check for existence of FTS records
if cerror <> 3143
  data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
file unload view "FTSG.vw"

'Send to INV group file
file load standard-view "INVG.vw"
data goto window 1
order change physical
data query execute "INV" index "INV"
'GROUP"="INV"
keys Enter,"INV",Enter,Esc
'Check for existence of INV records
if cerror <> 3143
  data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
file unload view "INVG.vw"

'Send to SB group file
file load standard-view "SBG.vw"
data goto window 1
order change physical
data query execute "SB" index "SB"
'GROUP"="SB"
keys Enter,"SB",Enter,Esc
'Check for existence of SB records
if cerror <> 3143
  data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
file unload view "SBG.vw"

'Send to OTHER group file
file load standard-view "OTHERG.vw"
data goto window 1
order change physical
data query execute "OTHER" index "OTHER"
'GROUP]="OTHER"
keys Enter,"OTHER",Enter,Esc
'Check for existence of OTHER records
if error <> 3143
data goto window 2
data utilities append "unidg.vw"
keys F10
else
data goto window 2
end if
goto window 2

else
file load standard-view "TALLEYX.vws"
data goto record first

[NETGL] = $netgl
[NETCB] = $newcb
[NETACH] = $newach
[NETCC] = $newcc
[NETCL] = $newcl
[NETCPN] = $newcpn
[NETFS] = $newfs
[NETFTS] = $newfts
[NETINV] = $newinv
[NETSB] = $newsb
[NETOTHER] = $newother
[NETUNID] = 0
[NETALL] = $newgl
print report execute "grpnogog.dfr" printer detail start 1 end 0 copies 1
file unload all
window close
jump out
end if
end if

label merge

screen clear 3 0
screen print 10 20 5 0 "Merging cumulative & current GL files "
screen draw box 8 18 12 59 3 0

data goto window 1
file load standard-view "cbg.vw"
data goto window 2
file load standard-view "cbgc.vws"
$cumcb = filesum([sign])
data utilities append "cbg.vw"
keys F10
if cerror = 3958
  $cuxcb = $cumcb
else
  $cuxcb = filesum([sign])
end if
if round($cuxcb,2) = round($cumcb + $newcb,2)
  data goto window 1
  file unload all
else
  open-printer
  lprint "Error in GL file merge routine for CB"
  lprint "Cumulative net activity prior to merge: " | str($cumcb)
  lprint "Net activity for today: " | str($newcb)
  lprint "New cumulative net activity: " | str($cuxcb)
  close-printer
  data goto window 1
  file unload all
end if

data goto window 1
file load standard-view "achg.vw"
data goto window 2
file load standard-view "achgc.vws"
$cumach = filesum([sign])
data utilities append "achg.vw"
keys F10
if cerror = 3958
  $cuxach = $cumach
else
  $cuxach = filesum([sign])
end if
if round($cuxach,2) = round($cumach + $newach,2)
  data goto window 1
  file unload all
else
  open-printer
  lprint "Error in GL file merge routine for group ACH"
  lprint "Cumulative net activity prior to merge: " | str($cumach)
  lprint "Net activity for today: " | str($newach)
  lprint "New cumulative net activity: " | str($cuxach)
  close-printer
  data goto window 1
  file unload all
end if

data goto window 1
file load standard-view "ccg.vw"
data goto window 2
file load standard-view "ccgc.vws"
$cumcc = filesum([sign])
data utilities append "ccg.vw"
keys F10
if cerror = 3958
  $cuxcc = $cumcc
else
  $cuxcc = filesum([sign])
end if
if round($cuxcc, 2) = round($cumcc + $newcc, 2)
    data goto window 1
    file unload all
else
    open-printer
    lprint "Error in GL file merge routine for group CC"
    lprint "Cumulative net activity prior to merge: " | str($cumcc)
    lprint "Net activity for today: " | str($newcc)
    lprint "New cumulative net activity: " | str($cuxcc)
    close-printer
    data goto window 1
    file unload all
end if

data goto window 1
file load standard-view "clg.vw"
data goto window 2
file load standard-view "clgc.vws"
$cumcl = filesum([sign])
data utilities append "clg.vw"
keys F10
if cerror = 3958
    $cuxcl = $cumcl
else
    $cuxcl = filesum([sign])
end if
if round($cuxcl, 2) = round($cumcl + $newcl, 2)
    data goto window 1
    file unload all
else
    open-printer
    lprint "Error in GL file merge routine for group CL"
    lprint "Cumulative net activity prior to merge: " | str($cumcl)
    lprint "Net activity for today: " | str($newcl)
    lprint "New cumulative net activity: " | str($cuxcl)
    close-printer
    data goto window 1
    file unload all
end if

data goto window 1
file load standard-view "cpng.vw"
data goto window 2
file load standard-view "cpngc.vws"
$cumcpn = filesum([sign])
data utilities append "cpng.vw"
keys F10
if cerror = 3958
    $cuxcpn = $cumcpn
else
    $cuxcpn = filesum([sign])
end if
if round($cuxcpn, 2) = round($cumcpn + $newcpn, 2)
    data goto window 1
    file unload all
else

open-printer
lprint "Error in GL file merge routine for group CPN"
lprint "Cumulative net activity prior to merge: "|str($cumcpn)
lprint "Net activity for today: "|str($newcpn)
lprint "New cumulative net activity: "|str($cuxcpn)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "fsg.vw"
data goto window 2
file load standard-view "fsgc.vws"
$cumfs = filesum([sign])
data utilities append "fsg.vw"
keys F10
if cerror = 3958
$cuxfs = $cumfs
else
$cuxfs = filesum([sign])
end if
if round($cuxfs,2) = round($cumfs + $newfs,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in GL file merge routine for group FS"
lprint "Cumulative net activity prior to merge: "|str($cumfs)
lprint "Net activity for today: "|str($newfs)
lprint "New cumulative net activity: "|str($cuxfs)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "ftsg.vw"
data goto window 2
file load standard-view "ftsgc.vws"
$cumfts = filesum([sign])
data utilities append "ftsg.vw"
keys F10
if cerror = 3958
$cuxfts = $cumfts
else
$cuxfts = filesum([sign])
end if
if round($cuxfts,2) = round($cumfts + $newfts,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in GL file merge routine for group FTS"
lprint "Cumulative net activity prior to merge: "|str($cumfts)
lprint "Net activity for today:" | str($newfts)
lprint "New cumulative net activity:" | str($cuxfts)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "invg.vw"
data goto window 2
file load standard-view "invgc.vws"
$cuminv = filesum([sign])
data utilities append "invg.vw"
keys F10
if cerror = 3958
$cuxinv = $cuminv
else
$cuxinv = filesum([sign])
end if
if round($cuxinv,2) = round($cuminv + $newinv,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in GL file merge routine for group INV"
lprint "Cumulative net activity prior to merge: " | str($cuminv)
lprint "Net activity for today: " | str($newinv)
lprint "New cumulative net activity: " | str($cuxinv)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "stg.vw"
data goto window 2
file load standard-view "stgc.vws"
$cumsb = filesum([sign])
data utilities append "stg.vw"
keys F10
if cerror = 3958
$cuxsb = $cumsb
else
$cuxsb = filesum([sign])
end if
if round($cuxsb,2) = round($cumsb + $newsb,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in GL file merge routine for group SB"
lprint "Cumulative net activity prior to merge: " | str($cumsb)
lprint "Net activity for today: " | str($newsb)
lprint "New cumulative net activity: " | str($cuxsb)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "otherg.vw"
data goto window 2
file load standard-view "othergc.vws"
$cumother = filesum([sign])
data utilities append "otherg.vw"
keys F10
if error = 3958
$cuxother = $cumother
else
$cuxother = filesum([sign])
end if
if round($cuxother,2) = round($cumother + $newother,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in GL file merge routine for group OTHER"
lprint "Cumulative net activity prior to merge:  "|str($cumother)
lprint "Net activity for today:  "|str($newother)
lprint "New cumulative net activity:  "|str($cuxother)
close-printer
data goto window 1
file unload all
end if

window close

file load standard-view "talleyg.vws"
data goto record first
$netgl = $cuxcb+$cuxach+$cuxcc+$cuxcl+$cuxcpn+$cuxfs+$cuxfts+$cuxinv+\$cuxother+$cuxsb
[NETGL] = $netgl
[NETALL] = $netgl

[NETCB] = $cuxcb
[NETACH] = $cuxach
[NETCC] = $cuxcc
[NETCL] = $cuxcl
[NETCPN] = $cuxcpn
[NETFS] = $cuxfs
[NETFTS] = $cuxfts
[NETINV] = $cuxinv
[NETOTHER] = $cuxother
[NETSB] = $cuxsb
file unload all

'Erase all of today's group files so that they cannot be reused

tools file erase "CBG.DB"
tools file erase "CBG.VW"
tools file erase "CBG.KEY"
tools file erase "ACHG.DB"
tools file erase "ACHG.VW"
tools file erase "ACHG.KEY"
tools file erase "CCG.DB"
tools file erase "CCG.VW"
tools file erase "CCG.KEY"
tools file erase "CLG.DB"
tools file erase "CLG.VW"
tools file erase "CLG.KEY"
tools file erase "CPNG.DB"
tools file erase "CPNG.VW"
tools file erase "CPNG.KEY"
tools file erase "FSG.DB"
tools file erase "FSG.VW"
tools file erase "FSG.KEY"
tools file erase "FTSG.DB"
tools file erase "FTSG.VW"
tools file erase "FTSG.KEY"
tools file erase "INVG.DB"
tools file erase "INVG.VW"
tools file erase "INVG.KEY"
tools file erase "SBG.DB"
tools file erase "SBG.VW"
tools file erase "SBG.KEY"
tools file erase "OTHERG.DB"
tools file erase "OTHERG.VW"
tools file erase "OTHERG.KEY"
tools file erase "UNIDG.DB"
tools file erase "UNIDG.VW"
tools file erase "UNIDG.KEY"

label out
end main
ASSIGNPF.PF3

'Variable definition
global $addcb $adch $addcc $addcl $addcpe $addfs
global $adfts $addinv $addsb $adother
global $newcb $newch $newcc $newcl $newcpe $newfs
global $newfts $newinv $newsb $newother
global $newgl $newql
global $cumcb $cuxcb $cumach $cuxach $cumcc $cuxcc $cumcl $cuxcl
global $cumcpe $cuxcpe $cumfs $cuxfs $cumfts $cuxfts $cuminv $cuxinv

main

repa int off
screen clear 3 0
screen print 10 20 5 0 "Please wait . . ."
screen draw box 8 18 12 59 3 0

' Check if unidentified items file exists, if it does not then this routine has already been run for today and should not be run again.

if file("c:\auto\unidf.db") = 0
screen clear 3 0
screen print 10 20 3 0 "Unidentified FRB items have already been assigned"
wait 2.5
jump out
else

' Load unidentified items file, bring onto screen for group code assignment
file load standard-view "unidf.vw"
order sort execute "debcre.dfs" index "debcre.idx"
repa int on

data enter
repa int off
screen clear 3 0
screen print 10 20 5 0 "Please wait . . ."
screen draw box 8 18 12 59 3 0

' Unassigned items are given code OTHER

data query execute "nogroup.dfq"

'([GROUP]="CB" AND [GROUP]="ACH" AND [GROUP]="CC" AND [GROUP]="CL"
' AND [GROUP]="CPN" AND [GROUP]="FS" AND [GROUP]="FTS" AND
' [GROUP]="INV" AND [GROUP]="SB" AND [GROUP]="OTHER")
'REPLACE [GROUP]="OTHER"
keys Esc

' Calculate net activity for each group

data query execute "CB" index "CB"
keys Esc
if cerror <> 3143
$addcb = filesum([sign])
else
$addcb = 0
end if
order change physical

data query execute "ACH" index "ACH"
keys Esc
if cerror <> 3143
    $addach = filesum(\{sign\})
else
    $addach = 0
end if
order change physical

data query execute "CC" index "CC"
keys Esc
if cerror <> 3143
    $addcc = filesum(\{sign\})
else
    $addcc = 0
end if
order change physical

data query execute "CL" index "CL"
keys Esc
if cerror <> 3143
    $addcl = filesum(\{sign\})
else
    $addcl = 0
end if
order change physical

data query execute "CPN" index "CPN"
keys Esc
if cerror <> 3143
    $addcpn = filesum(\{sign\})
else
    $addcpn = 0
end if
order change physical

data query execute "FS" index "FS"
keys Esc
if cerror <> 3143
    $addfs = filesum(\{sign\})
else
    $addfs = 0
end if
order change physical

data query execute "FTS" index "FTS"
keys Esc
if cerror <> 3143
$addfts = filesum([sign])
else
    $addfts = 0
end if
order change physical

data query execute "INV" index "INV"
keys Esc
if cerror <> 3143
    $addinv = filesum([sign])
else
    $addinv = 0
end if
order change physical

data query execute "SB" index "SB"
keys Esc
if cerror <> 3143
    $addsb = filesum([sign])
else
    $addsb = 0
end if
order change physical

data query execute "OTHER" index "OTHER"
keys Esc
if cerror <> 3143
    $addother = filesum([sign])
else
    $addother = 0
end if
order change physical

'Add new group activity to previous group activity

file load standard-view "TALLEYF.vws"
data goto record first
$netgl = [NETGL]
$newcb = $addcb + [NETCB]
$newach = $addach + [NETACH]
$newcc = $addcc + [NETCC]
$newcl = $addcl + [NETCL]
$newcpn = $addcpn + [NETCPN]
$newfs = $addfs + [NETFS]
$newfts = $addfts + [NETFTS]
$newinv = $addinv + [NETINV]
$newsb = $addsb + [NETSB]
$newother = $addother + [NETOTHER]

$newgl =$newcb+$newach+$newcc+$newcl+$newcpn+$newfs+$newfts+$newinv+$newsb+$newother

'If net totals agree, print report and send records to group files
if round($newgl,2) = round($netgl,2)
data goto record first
[NETCB] = $newcb
[NETACH] = $newach
[NETCC] = $newcc
[NETCL] = $newcl
[NETCPN] = $newcpn
[NETFS] = $newfs
[NETFTS] = $newfts
[NETINV] = $newinv
[NETSB] = $newsb
[NETOTHER] = $newother
[NETUNID] = 0
[NETALL] = $newgl

print report execute "groupof.dfr" printer detail start 1 end 0 copies 1
data goto view "unidf.vw"
window split vertical 42
data goto window 2

'Send to CB group file
file load standard-view "CBF.vw"
data goto window 1
order change physical
data query execute "CB" index "CB"
'GROUP"="CB"
keys Enter,"CB",Enter,Esc
'Check for existence of CB records
if error <> 3143
data goto window 2
data utilities append "unidf.vw"
keys F10
else
data goto window 2
end if
file unload view "CBF.vw"

'Send to ACH group file
file load standard-view "ACHF.vw"
data goto window 1
order change physical
data query execute "ACH" index "ACH"
'GROUP"="ACH"
keys Enter,"ACH",Enter,Esc
'Check for existence of ACH records
if error <> 3143
data goto window 2
data utilities append "unidf.vw"
keys F10
else
data goto window 2
end if
file unload view "ACHF.vw"

'Send to CC group file
file load standard-view "CCF.vw"
data goto window 1
order change physical
data query execute "CC" index "CC"
'[GROUP]="CC"
keys Enter,"CC",Enter,Esc
'Check for existence of CC records
if cerror <> 3143
   data goto window 2
   data utilities append "unidf.vw"
   keys F10
else
   data goto window 2
end if
file unload view "CCP.vw"

'Send to CL group file
file load standard-view "CLF.vw"
data goto window 1
order change physical
data query execute "CL" index "CL"
'[GROUP]="CL"
keys Enter,"CL",Enter,Esc
'Check for existence of CL records
if cerror <> 3143
   data goto window 2
   data utilities append "unidf.vw"
   keys F10
else
   data goto window 2
end if
file unload view "CLF.vw"

'Send to CPN group file
file load standard-view "CPNF.vw"
data goto window 1
order change physical
data query execute "CPN" index "CPN"
'[GROUP]="CPN"
keys Enter,"CPN",Enter,Esc
'Check for existence of CPN group records
if cerror <> 3143
   data goto window 2
   data utilities append "unidf.vw"
   keys F10
else
   data goto window 2
end if
file unload view "CPNF.vw"

'Send to FS group file
file load standard-view "FSF.vw"
data goto window 1
order change physical
data query execute "FS" index "FS"
'[GROUP]="FS"
keys Enter,"FS",Enter,Esc
'Check for existence of FS records
if cerror <> 3143
data goto window 2  
data utilities append "unidf.vw"  
keys F10  
else  
data goto window 2  
end if  
file unload view "FSP.vw"  

'Send to FTS group file  
file load standard-view "FTSF.vw"  
data goto window 1  
order change physical  
data query execute "FTS" index "FTS"  
'[GROUP]="FTS"  
keys Enter,"FTS",Enter,Esc  
'Check for existence of FTS records  
if cerror <> 3143  
data goto window 2  
data utilities append "unidf.vw"  
keys F10  
else  
data goto window 2  
end if  
file unload view "FTSF.vw"  

'Send to INV group file  
file load standard-view "INVF.vw"  
data goto window 1  
order change physical  
data query execute "INV" index "INV"  
'[GROUP]="INV"  
keys Enter,"INV",Enter,Esc  
'Check for existence of INV records  
if cerror <> 3143  
data goto window 2  
data utilities append "unidf.vw"  
keys F10  
else  
data goto window 2  
end if  
file unload view "INVF.vw"  

'Send to SB group file  
file load standard-view "SBF.vw"  
data goto window 1  
order change physical  
data query execute "SB" index "SB"  
'[GROUP]="SB"  
keys Enter,"SB",Enter,Esc  
'Check for existence of SB records  
if cerror <> 3143  
data goto window 2  
data utilities append "unidf.vw"  
keys F10  
else  
data goto window 2  
end if  
file unload view "SBF.vw"
'Send to OTHER group file
file load standard-view "OTHERF.vw"
data goto window 1
order change physical
data query execute "OTHER" index "OTHER"
'[GROUP]="OTHER"
keys Enter,"OTHER",Enter,Esc
'Check for existence of OTHER records
if cerror <> 3143
data goto window 2
data utilities append "unidf.vw"
keys F10
else
data goto window 2
end if
jump merge

else
file load standard-view "TALLEYX.vws"
data goto record first
[NETGL] = $netgl
[NETCB] = $newcb
[NETACH] = $newach
[NETCC] = $newcc
[NETCL] = $newcl
[NETCPN] = $newcpn
[NETFS] = $newfs
[NETFTS] = $newfts
[NETINV] = $newinv
[NETSB] = $newsb
[NETOTHER] = $newother
[NETUNID] = 0
[NETALL] = $newgl
print report execute "grpnogof.dfr" printer detail start 1 end 0 copies 1
file unload all
window close
jump out
end if
end if

label merge

screen clear 3 0
screen print 10 20 5 0 "Merging cumulative & current FRB files"
screen draw box 8 18 12 59 3 0

data goto window 1
file load standard-view "cbf.vw"
data goto window 2
file load standard-view "cbfc.vws"
$cumcb = filesum({sign})
data utilities append "cbf.vw"
keys F10
if cerror = 3958
$cuxcb = $cumcb
else
$cuxcb = filesum([sign])
end if
if round($cuxcb,2) = round($cumcb + $newcb,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in FRB file merge routine for CB"
lprint "Cumulative net activity prior to merge: "|str($cumcb)
lprint "Net activity for today: "|str($newcb)
lprint "New cumulative net activity: "|str($cuxcb)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "achf.vw"
data goto window 2
file load standard-view "achfc.vws"
$cumach = filesum([sign])
data utilities append "achf.vw"
keys F10
if cerror = 3958
$cuxach = $cumach
else
$cuxach = filesum([sign])
end if
if round($cuxach,2) = round($cumach + $newach,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in FRB file merge routine for group ACH"
lprint "Cumulative net activity prior to merge: "|str($cumach)
lprint "Net activity for today: "|str($newach)
lprint "New cumulative net activity: "|str($cuxach)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "ccf.vw"
data goto window 2
file load standard-view "ccfc.vws"
$cumcc = filesum([sign])
data utilities append "ccf.vw"
keys F10
if cerror = 3958
$cuxcc = $cumcc
else
$cuxcc = filesum([sign])
end if
if round($cuxcc,2) = round($cumcc + $newcc,2)
data goto window 1
file unload all
else
  open-printer
  lprint "Error in FRB file merge routine for group CC"
  lprint "Cumulative net activity prior to merge: \"str($cumcc)"
  lprint "Net activity for today: \"str($newcc)"
  lprint "New cumulative net activity: \"str($cuxcc)"
  close-printer
  data goto window 1
  file unload all
end if

data goto window 1
file load standard-view "clf.vw"
data goto window 2
file load standard-view "clfc.vws"
$cumcl = filesum([sign])
data utilities append "clf.vw"
keys F10
if cerror = 3958
  $cuxcl = $cumcl
else
  $cuxcl = filesum([sign])
end if
if round($cuxcl,2) = round($cumcl + $newcl,2)
  data goto window 1
  file unload all
else
  open-printer
  lprint "Error in FRB file merge routine for group CL"
  lprint "Cumulative net activity prior to merge: \"str($cumcl)"
  lprint "Net activity for today: \"str($newcl)"
  lprint "New cumulative net activity: \"str($cuxcl)"
  close-printer
  data goto window 1
  file unload all
end if

data goto window 1
file load standard-view "cpnf.vw"
data goto window 2
file load standard-view "cpnfc.vws"
$cumcpn = filesum([sign])
data utilities append "cpnf.vw"
keys F10
if cerror = 3958
  $cuxcpn = $cumcpn
else
  $cuxcpn = filesum([sign])
end if
if round($cuxcpn,2) = round($cumcpn + $newcpn,2)
  data goto window 1
  file unload all
else
  open-printer
  lprint "Error in FRB file merge routine for group CPN"
  lprint "Cumulative net activity prior to merge: \"str($cumcpn)"
lprint "Net activity for today: \n  New cumulative net activity: \nend if

data goto window 1
file unload all
end if

data goto window 1
file load standard-view "fsf.vw"
data goto window 2
file load standard-view "fsfc.vws"
$cumfs = filesum([sign])
data utilities append "fsf.vw"
keys F10
if cerror = 3958
$cxuft = $cumfs
else
$cxuft = filesum([sign])
end if
if round($cxuft,2) = round($cumfs + $newft,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in FRB file merge routine for group FS"
lprint "Cumulative net activity prior to merge: ":str($cumfs)
lprint "Net activity for today: ":str($newft)
lprint "New cumulative net activity: ":str($cxuft)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "ftsf.vw"
data goto window 2
file load standard-view "ftsfc.vws"
$cumfts = filesum([sign])
data utilities append "ftsf.vw"
keys F10
if cerror = 3958
$cxuft = $cumfts
else
$cxuft = filesum([sign])
end if
if round($cxuft,2) = round($cumfts + $newfts,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in FRB file merge routine for group FTPS"
lprint "Cumulative net activity prior to merge: ":str($cumfts)
lprint "Net activity for today: ":str($newfts)
lprint "New cumulative net activity: ":str($cxuft)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "invf.vw"
data goto window 2
file load standard-view "invfc.vws"
$cuminv = filesum([sign])
data utilities append "invf.vw"
keys F10
if cerror = 3958
$cuxinv = $cuminv
else
$cuxinv = filesum([sign])
end if
if round($cuxinv,2) = round($cuminv + $newinv,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in FRB file merge routine for group INV"
lprint "Cumulative net activity prior to merge:  ":str($cuminv)
lprint "Net activity for today:  ":str($newinv)
lprint "New cumulative net activity:  ":str($cuxinv)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "sbf.vw"
data goto window 2
file load standard-view "sbfc.vws"
$cusmb = filesum([sign])
data utilities append "sbf.vw"
keys F10
if cerror = 3958
$cuxsb = $cusmb
else
$cuxsb = filesum([sign])
end if
if round($cuxsb,2) = round($cusmb + $newsb,2)
data goto window 1
file unload all
else
open-printer
lprint "Error in FRB file merge routine for group SB"
lprint "Cumulative net activity prior to merge:  ":str($cusmb)
lprint "Net activity for today:  ":str($newsb)
lprint "New cumulative net activity:  ":str($cuxsb)
close-printer
data goto window 1
file unload all
end if

data goto window 1
file load standard-view "otherf.vw"
data goto window 2
file load standard-view "otherfc.vws"
$cumother = $files($sign)
data utilities append "otherf.vw"
keys P10
if $error = 3958
    $cuxother = $cumother
else
    $cuxother = $files($sign)
end if
if round($cuxother,2) = round($cumother + $newother,2)
    data goto window 1
    file unload all
else
    open-printer
    lprint "Error in FRB file merge routine for group OTHER"
    lprint "Cumulative net activity prior to merge: "|str($cumother)
    lprint "Net activity for today: "|str($newother)
    lprint "New cumulative net activity: "|str($cuxother)
close-printer
data goto window 1
file unload all
end if

close-printer
file load standard-view "talleyf.vws"
data goto record first

$netgl = $cuxcb+$cuxach+$cuxcc+$cuxcl+$cuxcpn+$cuxft+$cuxinv+$cuxsb

[NETGL] = $netgl
[NETALL] = $netgl

[NETCB] = $cuxcb
[NETACH] = $cuxach
[NETCC] = $cuxcc
[NETCL] = $cuxcl
[NETCPN] = $cuxcpn
[NETFS] = $cuxfs
[NETFTS] = $cuxft
[NETINV] = $cuxinv
[NETOTHER] = $cuxother
[NETSB] = $cuxsb

file unload all

'Erase all of today's group files so that they cannot be reused

tools file erase "CBF.DB"
tools file erase "CBF.VW"
tools file erase "CBF.KEY"
tools file erase "ACHF.DB"
tools file erase "ACHF.VW"
tools file erase "ACHF.KEY"
tools file erase "CCF.DB"
tools file erase "CCF.VW"
tools file erase "CCF.KEY"
tools file erase "CLF.DB"
tools file erase "CLF.VW"
tools file erase "CLF.KEY"
tools file erase "CPNF.DB"
tools file erase "CPNF.VW"
tools file erase "CPNF.KEY"
tools file erase "FSF.DB"
tools file erase "FSF.VW"
tools file erase "FSF.KEY"
tools file erase "FTSF.DB"
tools file erase "FTSF.VW"
tools file erase "FTSF.KEY"
tools file erase "INVF.DB"
tools file erase "INVF.VW"
tools file erase "INVF.KEY"
tools file erase "SBF.DB"
tools file erase "SBF.VW"
tools file erase "SBF.KEY"
tools file erase "OTHERF.DB"
tools file erase "OTHERF.VW"
tools file erase "OTHERF.KEY"
tools file erase "UNIDF.DB"
tools file erase "UNIDF.VW"
tools file erase "UNIDF.KEY"

label out
end main
global $bchoice
main
file unload all
label menu
repaint on
screen clear box 1 1 21 80 0 5
screen clear box 2 14 20 67 0 0

screen draw box 2 14 20 67 15 0
screen print 3 25 11 0 "GL Account 1060 Reconciliation"
screen print 4 30 11 0 "Modify Group Files"
screen print 6 22 3 0 " GL " FRB
screen print 7 22 3 0 " == 
screen print 8 22 3 0 "A) CB"
screen print 9 22 3 0 "B) ACH"
screen print 10 22 3 0 "C) CC"
screen print 11 22 3 0 "D) CL"
screen print 12 22 3 0 "E) CPN"
screen print 13 22 3 0 "F) FS"
screen print 14 22 3 0 "G) FTS"
screen print 15 22 3 0 "H) INV"
screen print 16 22 3 0 "I) SB"
screen print 17 22 3 0 "J) OTHER"

screen print 8 45 3 0 "K) CB"
screen print 9 45 3 0 "L) ACH"
screen print 10 45 3 0 "M) CC"
screen print 11 45 3 0 "N) CL"
screen print 12 45 3 0 "O) CPN"
screen print 13 45 3 0 "P) FS"
screen print 14 45 3 0 "Q) FTS"
screen print 15 45 3 0 "R) INV"
screen print 16 45 3 0 "S) SB"
screen print 17 45 3 0 "T) OTHER"
screen print 18 28 3 0 "X) Return to Main Menu"
screen print 19 28 3 0 "Choice: "
screen input 19 36 3 0 1 $bchoice mask "[A-TXa-tx]"

case upper($bchoice)
  when "A"
    execute "cbgmod"
    jump menu
  when "B"
    execute "achgmod"
    jump menu
  when "C"
    execute "ccgmod"
    jump menu
  when "D"
    execute "clgmod"
    jump menu
  when "E"
    execute "cpngmod"
    jump menu
  when "F"
    execute "fsgmod"
jump menu
when "G"
  execute "itsgmod"
  jump menu
when "H"
  execute "invgmod"
  jump menu
when "I"
  execute "abgmod"
  jump menu
when "J"
  execute "othgmod"
  jump menu
when "K"
  execute "cbfmod"
  jump menu
when "L"
  execute "achfmod"
  jump menu
when "M"
  execute "ccfmod"
  jump menu
when "N"
  execute "clfmod"
  jump menu
when "O"
  execute "cpnfmod"
  jump menu
when "P"
  execute "fsfmod"
  jump menu
when "Q"
  execute "fssfmod"
  jump menu
when "R"
  execute "invimd"
  jump menu
when "S"
  execute "abfmod"
  jump menu
when "T"
  execute "othfmod"
  jump menu
when "X"
  transfer "menum"
otherwise
stop
end case
end main
CBGMOD.PF3

'Variable declaration
global $netach $netcb $netcc $netcl $netcpn $netfts $netinv $netsb
global $netother $netgl $netail $netunid
global $chgach $chgcdb $chgcoc $chgccl $chgcpcn $chgcfts $chgcinv $chgcgb
global $chgoother $chgall
global $oldach $oldcb $oldcc $oldcl $oldcpn $oldfts $oldinv $oldsb
global $oldother $oldall
global $group $diff $resp $deltot $addtot
main

'Clear screen
repaint off
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 30 3 0 "Please wait . . ."

$group = "CB"

file load standard-view "cbgc.vws"
$oldcb = filesum($group)

file load standard-view "achgc.vws"
$oldach = filesum($group)
file unload all

file load standard-view "ccgc.vws"
$oldcc = filesum($group)

file load standard-view "clgc.vws"
$oldcl = filesum($group)
file unload all

file load standard-view "cpngc.vws"
$oldcpn = filesum($group)

file load standard-view "fsgc.vws"
$oldfs = filesum($group)
file unload all

file load standard-view "ftsgc.vws"
$oldfts = filesum($group)

file load standard-view "invgc.vws"
$oldinv = filesum($group)
file unload all

file load standard-view "othergc.vws"
$oldother = filesum($group)

file load standard-view "sbgc.vws"
$oldsb = filesum($group)
file unload all

$oldall=$oldcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldfts+$oldfs+$oldfts\+$oldinv+$oldother+$oldsb
file load standard-view "talleyg.vws"
data goto record first
[NETCB] = $oldcb
[NETACH] = $oldach
[NETCC] = $oldcc
[NETCL] = $oldcl
[NETCPN] = $oldcpn
[NETFS] = $oldfs
[NETFTS] = $oldfts
[NETINV] = $oldinv
[NETOTHER] = $oldother
[NETSB] = $oldsb
[NETALL] = $oldcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldfs+$oldfts+$oldinv+
              $oldother+$oldsb
[NETGL] = $oldcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldfs+$oldfts+$oldinv+
              $oldother+$oldsb
file unload all

file load standard-view "cbgc.vws"
data utilities recalc-all
label enter
order sort execute "words.dfs" index "words.idx"
'Ascending amount within type (debit,credit)
repaint on
data enter
repaint off
screen clear 1 0
label replace
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 25 3 0 "Do you want to replace items? "
screen input 12 56 3 0 1 $resp
if $resp == "Y"
repaint on
data enter
repaint off
data query execute "replace" index "replace"
'[$REPLACE] == "R"
keys Esc
if cerror <> 3143
window split vertical 42
data goto window 2
file load standard-view "modg.vws"
data utilities append cbgc.vws
keys F10
data query execute "rcode"
'REPLACE [rcode] = "R"[str([RECORD])]
$deltot = filesum([sign])
label change
repaint on
data enter
repaint off
screen clear 1 0
data utilities recalc-all
Sadddtot = filesum([sign],[REPLACE] <> "R")
if round($deltot,2) = round($addtot,2)

screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 25 3 0 "Changes net to zero . . ."
data goto window 1
file load standard-view "modfileg.vws"
data utilities append "modg.vws"
keys F10
file unload view "modfileg.vws"
data goto view "cbgc.vws"
data query execute "delr"
'[REPLACE] = "R" REPLACE DELETE
data goto window 2
data query execute "added" index "added"
'[REPLACE] <> "R"
data goto window 1
data utilities append "modg.vws"
keys F10
file unload view "cbgc.vws"
data goto window 2
order change physical
data query execute "delall"
'REPLACE DELETE
keys Esc
file unload view "modg.vws"
window close
data utilities purge "modg.db"
data utilities purge "cbgc.db"
file load standard-view "cbgc.vws"
jump replace

'Changes do not net to zero
else
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 25 3 0 "Changes do not net to zero . . ."
wait 2
jump change
end if

'No records are selected for replacement
else
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 25 3 0 "No records were selected for replacement"
wait 2
jump replace
end if

'Do not want to replace records
else
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 30 3 0 "Please wait . . ."
end if

'Check for invalid group codes
data query execute "validg.dfq" index "validg.idx"

'[GROUP] <> "ACH" AND [GROUP] <> "CB" AND ...

keys Esc
if cerror = 3143
jump continue
else
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 17 3 0 "Invalid GROUP/RGROUP found, please correct"
wait 2
order change physical
jump enter
end if
label continue

'Recalculate [SIGN] in case [AMOUNT] changed
data utilities recalc-all

'Delete records with zero amount
data query execute "zero.dfq"

'[AMOUNT]=0
keys Esc

file unload view "cbgc.vws"
data utilities purge "cbgc.db"
file load standard-view "cbgc.vws"

'Check for group code changes
data query execute "notcb.dfq" index "notcb.idx"
keys Esc

'[GROUP] <> "CB"

'If no group code changes, check for amount changes
if cerror = 3143

$netcb = filesum({[sign]})
$chgc = round($netcb - $oldcb,2)
if $chgc = 0
file load standard-view "talleyg.vws"
data goto record first
[CHGACH] = 0
[CHGCB] = 0
[CHGCC] = 0
[CHGLC] = 0
[CHGCPN] = 0
[CHGF] = 0
[CHGFTS] = 0
[CHGINV] = 0
[CHGOTHER] = 0
[CHGSET] = 0
[CHGALL] = 0
[GROUP] = $group
$diff = [DIFERENCE]
if round($diff,2) = 0
print report execute "chggog.dfr" printer detail start 1 end 0

copies 1
else
print report execute "chnogog.dfr" printer detail start 1 end
0 copies 1
end if
else
file load standard-view "talleyg.vws"
data goto record first

[GROUP] = $group

$netall = $netcb+$oldach+$oldcc+$oldci+$oldcpn+$oldfs+
\soldfts+$oldinv+$oldother+$oldsb

$chgall = round($netall - $oldall,2)

[NETCB] = $netcb
[NETALL] = $netall
[CHGCB] = $chgcb
[CHGACH] = 0
[CHGCC] = 0
[CHGCL] = 0
[CHGCPN] = 0
[CHGFSS] = 0
[CHGFPS] = 0
[CHGINV] = 0
[CHGOTHER] = 0
[CHGSB] = 0
[CHGALL] = $chgall
$diff = [DIFERENCE]
if round($diff,2) = 0
print report execute "chggog.dfr" printer detail start 1 end 0

copies 1
else
print report execute "chnogog.dfr" printer detail start 1 end
0 copies 1
end if
end if

'If group codes changed, sent records to appropriate group files
else
window split vertical 42
data goto window 2

'Transfer records to ACH group file
file load standard-view "achgc.vws"
data goto window 1
order change physical
data query execute "ACH" index "ACH"

'[GROUP]="ACH"
keys Enter,"ACH",Enter,Esc
'Check for existence of ACH records
if error <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys F10
$netach = filesum([sign])
else
data goto window 2
$netach = filesum([sign])
end if
$chgach = round($netach - $oldach,2)
file unload view "achgc.vws"

' Transfer records to CC group file
file load standard-view "ccgc.vws"
data goto window 1
order change physical
data query execute "CC" index "CC"
'[GROUP]="CC"
keys Enter,"CC",Enter,Esc
' Check for existence of CC records
if cerror <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys F10
$netcc = filesum([sign])
else
data goto window 2
$netcc = filesum([sign])
end if
$chgcc = round($netcc - $oldcc,2)
file unload view "ccgc.vws"

' Transfer records to CL group file
file load standard-view "clgc.vws"
data goto window 1
order change physical
data query execute "CL" index "CL"
'[GROUP]="CL"
keys Enter,"CL",Enter,Esc
' Check for existence of CL records
if cerror <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys F10
$netcl = filesum([sign])
else
data goto window 2
$netcl = filesum([sign])
end if
$chgcl = round($netcl - $oldcl,2)
file unload view "clgc.vws"

' Transfer records to CPN group file
file load standard-view "cpngc.vws"
data goto window 1
order change physical
data query execute "CPN" index "CPN"
'[GROUP]="CPN"
keys Enter,"CPN",Enter,Esc
' Check for existence of CPN records
if cerror <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys F10
$netcnp = filesum([sign])
else
  data goto window 2
  $netcnp = filesum([sign])
end if
$chgcpn = round($netcnp - $oldcnp,2)
file unload view "cpngc.vws"

'Transfer records to FS group file
file load standard-view "fsgc.vws"
data goto window 1
order change physical
data query execute "FS" index "FS"
'[GROUP]="FS"
keys Enter,"FS",Enter,Esc
'Check for existence of FS records
if cerror <> 3143
  data goto window 2
  data utilities append "cbgc.vws"
  keys F10
  $netfs = filesum([sign])
else
  data goto window 2
  $netfs = filesum([sign])
end if
$chgsfs = round($netfs - $oldfs,2)
file unload view "fsgc.vws"

'Transfer records to FTS group file
file load standard-view "ftsfc.vws"
data goto window 1
order change physical
data query execute "FTS" index "FTS"
'[GROUP]="FTS"
keys Enter,"FTS",Enter,Esc
'Check for existence of FTS records
if cerror <> 3143
  data goto window 2
  data utilities append "cbgc.vws"
  keys F10
  $netfts = filesum([sign])
else
  data goto window 2
  $netfts = filesum([sign])
end if
$chgtfts = round($netfts - $oldfts,2)
file unload view "ftsfc.vws"

'Transfer records to INV group file
file load standard-view "invgc.vws"
data goto window 1
order change physical
data query execute "INV" index "INV"
' [GROUP]="INV"
keys Enter,"INV",Enter,Esc
' Check for existence of INV records
if cerror <> 3141
  data goto window 2
data utilities append "cbgc.vws"
keys F10
$netinv = filesum([sign])
else
data goto window 2
$netinv = filesum([sign])
end if
$chginv = round($netinv - $oldinv,2)
file unload view "invgc.vws"

' Transfer records to OTHER group file
file load standard-view "othergc.vws"
data goto window 1
order change physical
data query execute "OTHER" index "OTHER"
' [GROUP]="OTHER"
keys Enter,"OTHER",Enter,Esc
' Check for existence of OTHER records
if cerror <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys F10
$netother = filesum([sign])
else
data goto window 2
$netother = filesum([sign])
end if
$chgother = round($netother - $oldother,2)
file unload view "othergc.vws"

' Transfer records to SB group file
file load standard-view "sbgc.vws"
data goto window 1
order change physical
data query execute "SB" index "SB"
' [GROUP]="SB"
keys Enter,"SB",Enter,Esc
' Check for existence of SB records
if cerror <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys F10
$netsb = filesum([sign])
else
data goto window 2
$netsb = filesum([sign])
end if
$chgsb = round($netsb - $oldsb,2)
file unload view "sbgc.vws"
window close
data goto view "cbgc.vws"
'Delete records that went to other group files
data query execute "delcb"
keys Esc

'[GROUP] <> "CS" replace delete

file unload all
data utilities purge "cbgc.db"
file load standard-view "cbgc.vws"
$netcb = filesum([sign])
$chgc = round($netcb - $oldcb,2)
$netall = $netach+$netcb+$netcc+$netcl+$netcpn+$netsf$s
 +$netfts+$netinv+$netother+$netsb
$chgal = round($netall - $oldall,2)
file load standard-view "TALLEYG.vws"
data goto record first
$netgl = [*NETCL]

if round($netgl,2) = round($netall,2)
data goto record first
[NETACH] = $netach
[NETCB] = $netcb
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netsf$s
[NETFTS] = $netfts
[NETINV] = $netinv
[NETOTHER] = $netother
[NETSB] = $netsb
[NETALL] = $netall
[CHGACH] = $chgach
[CHGCB] = $chgc
[CHGCC] = $chgc
[CHGCL] = $chgl
[CHGCPN] = $chgc
[CHGFS] = $chgfs
[CHGFTS] = $chgfts
[CHGINV] = $chginv
[CHGOTHER] = $chgother
[CHGSB] = $chgsb
[CHGALL] = $chgal
[GROUP] = $group
print report execute "chggog.dfr" printer detail start 1 end 0 copies 1
else
[GROUP] = $group

[NETACH] = $netach
[NETCB] = $netcb
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netfs
[NETFTS] = $netfts
[NETINV] = $netinv
[NETOTHER] = $netother
[NETSB] = $netsb
[NETALL] = $netall

[CHGACH] = $chgach
[CHGCB] = $chgcbb
[CHGCC] = $chgccc
[CHGCL] = $chgccl
[CHGCPN] = $chgcpcn
[CHGFS] = $chgfs
[CHGFTS] = $chgftts
[CHGINV] = $chginv
[CHGOTHER] = $chgother
[CHGSB] = $chgsb
[CHGALL] = $chgall

print report execute "chgnogog.dfr" printer detail start 1 end 0 copies 1

end if
end if
file unload all
end main
CBFMOD.PF3

'Variable declaration
global $netach $netcb $netcc $netcl $netcpn $netftp $netinv $netsb
global $netother $netgl $netall $netunid
global $chgacn $chgcbl $chgccl $chgcpln $chgf $chgs $chginv $chgsb
global $chgoother $chgoall
global $oldach $oldcb $oldcc $oldcl $oldcpn $oldftp $oldinv $oldsb
global $oldother $oldall
global $group $diff $resp $deltot $addtot
main

'Clear screen
re paint off
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 30 3 0 "Please wait . . ."

$group = "CB"

file load standard-view "cbfc.vws"
$oldcb = filesum([sign])

file load standard-view "achfc.vws"
$oldach = filesum([sign])
file unload all

file load standard-view "ccfc.vws"
$oldcc = filesum([sign])

file load standard-view "clfc.vws"
$oldcl = filesum([sign])
file unload all

file load standard-view "cpnfc.vws"
$oldcpn = filesum([sign])

file load standard-view "fsfc.vws"
$oldfps = filesum([sign])
file unload all

file load standard-view "ftsfc.vws"
$oldfts = filesum([sign])

file load standard-view "invfc.vws"
$oldinv = filesum([sign])
file unload all

file load standard-view "otherfc.vws"
$oldother = filesum([sign])

file load standard-view "sbfc.vws"
$oldsb = filesum([sign])
file unload all

$oldall=$oldcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldftp+$oldinv+$oldsb
file load standard-view "talleyf.vws"
data goto record first
[NETCB] = $oldcb
[NETACH] = $oldach
[NETCC] = $oldcc
[NETCL] = $oldcl
[NETCPN] = $oldcpn
[NETPS] = $oldfs
[NETFTS] = $oldfts
[NETINV] = $oldinv
[NETOTHER] = $oldother
[NETSB] = $oldsb
[NETALL] = $oldcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldfs+$oldfts+$oldinv+\$oldother+$oldsb
[NETGL] = $oldcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldfs+$oldfts+$oldinv+\$oldother+$oldsb
file unload all

data unload all

file load standard-view "cbfc.vws"
data utilities recalc-all
label enter
order sort execute "wordx.dfs" index "wordx.idx"
'Ascending amount within type (debit,credit)'
repaint on
data enter

repa int off
screen clear 1 0
label replace
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 25 3 0 "Do you want to replace items? "
screen input 12 56 3 0 1 $resp

if $resp == "Y"
repaint on
data enter
repa int off
data query execute "replace" index "replace"
'[REPLACE] == "R"
keys Esc
if error <> 3143
window split vertical 42
data goto window 2
file load standard-view "modf.vws"
data utilities append "cbfc.vws"
keys F10
data query execute "rcode"
'REPLACE [RCODE] = "R"|STR([RECORD])
$deltot = filesum([sign])
label change
repa int on
data enter
repa int off
screen clear 1 0
data utilities recalc-all
$addtot = filesum([sign],[REPLACE] <> "R")
if round($deltot,2) = round($addtot,2)

    screen clear 1 0
    screen draw box 10 10 14 70 5 0
    screen print 12 25 3 0 "Changes net to zero . . ."
    data goto window 1
    file load standard-view "modfile.vws"
    data utilities append "modf.vws"
    keys F10
    file unload view "modfile.vws"
    data goto view "cbfc.vws"
    data query execute "delt"
    '[REPLACE] = "R" REPLACE DELETE
    data goto window 2
    data query execute "added" index "added"
    '[REPLACE] <> "R"
    data goto window 1
    data utilities append "modf.vws"
    keys F10
    file unload view "cbfc.vws"
    data goto window 2
    order change physical
    data query execute "delall"
    'REPLACE DELETE
    keys Esc
    file unload view "modf.vws"
    window close
    data utilities purge "modf.db"
    data utilities purge "cbfc.db"
    file load standard-view "cbfc.vws"
    jump replace

    Changes do not net to zero
else
    screen clear 1 0
    screen draw box 10 10 14 70 5 0
    screen print 12 25 3 0 "Changes do not net to zero . . ."
    wait 2
    jump change

end if

'No records are selected for replacement
else
    screen clear 1 0
    screen draw box 10 10 14 70 5 0
    screen print 12 25 3 0 "No records were selected for replacement"
    wait 2
    jump replace

end if

'Do not want to replace records
else
    screen clear 1 0
    screen draw box 10 10 14 70 5 0
    screen print 12 30 3 0 "Please wait . . ."
end if

' Check for invalid group codes
data query execute "validg.dfq" index "validg.idx"

'[GROUP] <> "ACH" AND [GROUP] <> "CB" AND ...

keys Esc
if cerror = 3143
    jump continue
else
    screen clear 1 0
    screen draw box 10 10 14 70 5 0
    screen print 12 17 3 0 "Invalid GROUP/RGROUP found, please correct"
    wait 2
    order change physical
    jump enter
end if
label continue

'Recalculate [SIGN] in case [AMOUNT] changed
data utilities recalc-all

'Delete records with zero amount
data query execute "zero.dfq"
'[SIGN]=0
keys Esc

file unload view "cbfc.vws"
data utilities purge "cbfc.db"
file load standard-view "cbfc.vws"

' Check for group code changes
data query execute "notcb.dfq" index "notcb.idx"
keys Esc

'[GROUP] <> "CB"

' If no group code changes, check for amount changes
if cerror = 3143

    $netcb = filesum([sign])
    $chgb = round($netcb - $oldcb,2)
    if $chgb = 0
        file load standard-view "tallyf.vws"
data goto record first
        [CHGACH] = 0
        [CHGCB] = 0
        [CHGCC] = 0
        [CHGCL] = 0
        [CHGCPN] = 0
        [CHGFS] = 0
        [CHGFTS] = 0
        [CHGINV] = 0
        [CHGOTHER] = 0
        [CHGSB] = 0
        [CHGALL] = 0
        [GROUP] = $group
$diff = "DIFFERENCE"
if round($diff,2) = 0
   print report execute "chggof.dfr" printer detail start 1 end 0
else
   print report execute "chnogof.dfr" printer detail start 1 end 0
end if

file load standard-view "talleyf.vws"
data goto record first
[GROUP] = $group
$netall = $netcb+$oldach+$oldcc+$oldcl+$oldcpn+$oldfs+
         $oldfts+$oldinv+$oldother+$oldsb
$chgall = round($netall - $oldall,2)
[NETCB] = $netcb
[NETALL] = $netall
[CHGCB] = $chgcb
[CHGACH] = 0
[CHGCC] = 0
[CHGCL] = 0
[CHGCPN] = 0
[CHGFS] = 0
[CHGFTS] = 0
[CHGINV] = 0
[CHGOTHER] = 0
[CHGSB] = 0
[CHGALL] = $chgall
$diff = "DIFFERENCE"
if round($diff,2) = 0
   print report execute "chggof.dfr" printer detail start 1 end 0
else
   print report execute "chnogof.dfr" printer detail start 1 end 0
end if

'If group codes changed, sent records to appropriate group files
else
   window split vertical 42
data goto window 2

'Transfer records to ACH group file
file load standard-view "achfc.vws"
data goto window 1
order change physical
data query execute "ACH" index "ACH"
'[GROUP]="ACH"
keys Enter,"ACH",Enter,Esc
'Check for existence of ACH records
if error <> 3143
   data goto window 2
data utilities append "cbfc.vws"
keys F10
   $netach = filesum([sign])
else
data goto window 2
$netach = filesum([sign])
end if
$chgach = round($netach - $oldach,2)
file unload view "achfc.vws"

'Transfer records to CC group file
file load standard-view "ccfc.vws"
data goto window 1
order change physical
data query execute "CC" index "CC"
'[GROUP]="CC"
keys Enter,"CC",Enter,Esc
'Check for existence of CC records
if cerror <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys F10
$netcc = filesum([sign])
else
data goto window 2
$netcc = filesum([sign])
end if
$chgcc = round($netcc - Soldcc,2)
file unload view "ccfc.vws"

'Transfer records to CL group file
file load standard-view "clf.c.vws"
data goto window 1
order change physical
data query execute "CL" index "CL"
'[GROUP]="CL"
keys Enter,"CL",Enter,Esc
'Check for existence of CL records
if cerror <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys F10
$netcl = filesum([sign])
else
data goto window 2
$netcl = filesum([sign])
end if
$chgcl = round($netcl - Soldcl,2)
file unload view "clf.c.vws"

'Transfer records to CPN group file
file load standard-view "cpnf.c.vws"
data goto window 1
order change physical
data query execute "CPN" index "CPN"
'[GROUP]="CPN"
keys Enter,"CPN",Enter,Esc
'Check for existence of CPN records
if cerror <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys F10
$netcpn = filesum([sign])
else
data goto window 2
$netcpn = filesum([sign])
end if
$chgcpn = round($netcpn - $oldcpn,2)
file unload view "cpnfc.vws"

'Transfer records to FS group file
file load standard-view "fsfc.vws"
data goto window 1
order change physical
data query execute "FS" index "FS"
'[GROUP]="FS"
keys Enter,"FS",Enter,Esc
'Check for existence of FS records
if error <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys F10
$netfs = filesum([sign])
else
data goto window 2
$netfs = filesum([sign])
end if
$chgfs = round($netfs - $oldfs,2)
file unload view "fsfc.vws"

'Transfer records to FTPS group file
file load standard-view "ftsfc.vws"
data goto window 1
order change physical
data query execute "FTS" index "FTS"
'[GROUP]="FTS"
keys Enter,"FTS",Enter,Esc
'Check for existence of FTPS records
if error <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys F10
$netfts = filesum([sign])
else
data goto window 2
$netfts = filesum([sign])
end if
$chgfts = round($netfts - $oldfts,2)
file unload view "ftsfc.vws"

'Transfer records to INV group file
file load standard-view "invfc.vws"
data goto window 1
order change physical
data query execute "INV" index "INV"
'[GROUP]="INV"
keys Enter,"INV",Enter,Esc
'Check for existence of INV records
if cerror <> 3143
   data goto window 2
   data utilities append "cbfc.vws"
   keys F10
   $netinv = filesum([sign])
else
   data goto window 2
   $netinv = filesum([sign])
end if
$chginv = round($netinv - $oldinv,2)
file unload view "invfc.vws"

'Transfer records to OTHER group file
file load standard-view "otherfc.vws"
data goto window 1
order change physical
data query execute "OTHER" index "OTHER"
'[GROUP]="OTHER"
keys Enter,"OTHER",Enter,Esc
'Check for existence of OTHER records
if cerror <> 3143
   data goto window 2
   data utilities append "cbfc.vws"
   keys F10
   $netother = filesum([sign])
else
   data goto window 2
   $netother = filesum([sign])
end if
$chgother = round($netother - $oldother,2)
file unload view "otherfc.vws"

'Transfer records to SB group file
file load standard-view "sbfc.vws"
data goto window 1
order change physical
data query execute "SB" index "SB"
'[GROUP]="SB"
keys Enter,"SB",Enter,Esc
'Check for existence of SB records
if cerror <> 3143
   data goto window 2
   data utilities append "cbfc.vws"
   keys F10
   $netsb = filesum([sign])
else
   data goto window 2
   $netsb = filesum([sign])
end if
$chgsb = round($netsb - $oldsb,2)
file unload view "asfbc.vws"
window close
data goto view "cbfc.vws"
'Delete records that went to other group files
data query execute "delcb"
keys Esc

'[GROUP] <> "CB" replace delete

file unload all
data utilities purge "cbfc.db"
file load standard-view "cbfc.vws"
$netcb = filesum([sign])
$chgcbl = round($netcb - $oldcb,2)
$netall = $netach+$netcb+$netcc+$netcl+$netcpn+$netsb+
+netsfs+$netinv+$netother+$netsb
$chgal = round($netall - $oldall,2)
file load standard-view "talleyf.vws"
data goto record first
$netgl = [NETGL]

if round($netgl,2) = round($netall,2)
data goto record first
[NETACH] = $netach
[NETCB] = $netcb
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netsfs
[NETINV] = $netinv
[NETOTHER] = $netother
[NETSB] = $netsb
[NETALL] = $netall

[CHGACH] = $chgach
[CHGCB] = $chgcbl
[CHGCC] = $chgcbl
[CHGCL] = $chgcbl
[CHGCOPN] = $chgcpl
[CHGFSS] = $chgsfs
[CHGFSTS] = $chgfts
[CHGINV] = $chginv
[CHGOTHER] = $chgother
[CHGSB] = $chgsb
[CHGALL] = $chgal
[GROUP] = $group
print report execute "chggof.dfr" printer detail start 1 end 0 copies 1

else
[GROUP] = $group

[NETACH] = $netach
[NETCB] = $netcb
[NETCC] = $netcc
[NETCL] = $netcl
[NETCPN] = $netcpn
[NETFS] = $netfs
[NETFTS] = $netfts
[NETINV] = $netinv
[NETOTHER] = $netother
[NETSB] = $netsb
[NETALL] = $netall

[CHGACH] = $chgach
[CHGCB] = $chgcb
[CHGCC] = $chgcc
[CHGCL] = $chgcl
[CHGCPN] = $chgcpn
[CHGFS] = $chqfs
[CHGFTS] = $chqfts
[CHGINV] = $chqinv
[CHGOTHER] = $chgother
[CHGSB] = $chgsb
[CHGALL] = $chgall

print report execute "chgnogof.dfr" printer detail start 1 end 0 copies 1
end if
end if
file unload all
end main
global $achoice
main
file unload all
label menu
repaint on
screen clear box 1 1 21 80 0 5
screen clear box 2 14 20 67 0 0

screen draw box 2 14 20 67 15 0
screen print 4 25 11 0 "GL Account 1060 Reconciliation"
screen print 5 22 11 0 "Automated Matching Function by Group"
screen print 6 22 11 0 "=====================================
screen print 7 25 3 0 "A) Correspondent Banks (CB)"
screen print 8 25 3 0 "B) Automated Clearing House (ACH)"
screen print 9 25 3 0 "C) Coin & Currency (CC)"
screen print 10 25 3 0 "D) Cash Letters (CL)"
screen print 11 25 3 0 "E) Coupon Payments (CPN)"
screen print 12 25 3 0 "F) Food Stamps (FS)"
screen print 13 25 3 0 "G) Funds Transfer (FTS)"
screen print 14 25 3 0 "H) Investments (INV)"
screen print 15 25 3 0 "I) OTHER"
screen print 16 25 3 0 "J) Savings Bonds (SB)"
screen print 17 25 3 0 "X) Return to Main Menu"
screen print 19 25 3 0 "Choice:"

screen input 19 33 3 0 1 $achoice mask "[A-Xa-jx]"

case upper($achoice)
when "A"
execute "cbm"
jump menu
when "B"
execute "achm"
jump menu
when "C"
execute "ccm"
jump menu
when "D"
execute "clm"
jump menu
when "E"
execute "cpnm"
jump menu
when "F"
execute "fsm"
jump menu
when "G"
execute "ftsm"
jump menu
when "H"
execute "invm"
jump menu
when "I"
execute "otherm"
jump menu
when "J"
execute "sbm"
jump menu
when "X"
   transfer "menu"
otherwise
   jump menu
end case
end main
CBM.PF3

public $gcode

main

repaing off
screen clear box 1 1 21 80 0 5
screen clear box 10 14 14 67 0 0
screen draw box 10 14 14 67 15 0
screen print 12 20 15 0 "Matching in process, please wait . . ."
$gcode = "CB"

window split vertical 37
file load standard-view "cbgc.vws"
order change key "[DEBCRE]"

data goto window 2
file load standard-view "cbfc.vws"
order change key "[DEBCRE]"

data goto window 1
tools file erase "cbmatch.*"

data relate execute "cb.dfx"

data transact execute "cbf.dft"
data goto window 2
data goto view "cbgc.vws"

data goto window 1
data transact execute "cbg.dft"

file unload view "cbmatch.vws"

file load standard-view "MATCHES"
data goto window 2
data query execute "MATCHES" index "MATCHES"
keys Enter, "MATCHES", Enter, Esc
'LEFT([MATCH],1) = "A"
if error <> 3143
data goto window 1
data utilities append "cbgc.vws"
keysDown, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter, F7, Right
keys Right, Right, Down, Enter, F10
data goto window 2
else
data goto window 2
end if
order change physical
data goto view "cbfc.vws"
data query execute "MATCHES" index "MATCHES"
keys Enter, "MATCHES", Enter, Esc
'LEFT([MATCH],1) = "A"
if error <> 3143
data goto window 1
data utilities append "cbfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right, Right, Right
keys Enter, Down, F7, Down, Right, Right, Enter, F10
else
screen print 12 20 15 0 "No automatic matches made"
wait 2.5
file unload all
jump end
end if

order change key "[AMOUNT]"
print report execute "matches" printer detail start 1 end 0 copies 1
data query execute "delall"
'REPLACE DELETE
keys Esc
file unload all
data utilities purge "matches.db"
label end
window close
repaint on
MENUCLR.PF1

global $cchoice
main
file unload all
label menu
repaint on
screen clear box 1 1 21 80 0 5
screen clear box 2 14 20 67 0 0

screen draw box 2 14 20 67 15 0
screen print 4 25 11 0 "GL Account 1060 Reconciliation"
screen print 5 30 11 0 "Clearing by Group"
screen print 6 22 11 0 "==================================
screen print 8 25 3 0 "A) Correspondent Banks (CB)"
screen print 9 25 3 0 "B) Automated Clearing House (ACH)"
screen print 10 25 3 0 "C) Coin & Currency (CC)"
screen print 11 25 3 0 "D) Cash Letters (CL)"
screen print 12 25 3 0 "E) Coupon Payments (CPN)"
screen print 13 25 3 0 "F) Food Stamps (FS)"
screen print 14 25 3 0 "G) Funds Transfer System (FTS)"
screen print 15 25 3 0 "H) Investments (INV)"
screen print 16 25 3 0 "I) OTHER"
screen print 17 25 3 0 "J) Savings Bonds (SB)"
screen print 18 25 3 0 "X) Return to Main Menu"
screen print 19 25 3 0 "Choice: 
screen input 19 33 3 0 1 $cchoice mask "{A-JXa-jx}"
case upper($cchoice)
when "A"
execute "cbclear"
jump menu
when "B"
execute "achclear"
jump menu
when "C"
execute "cccclear"
jump menu
when "D"
execute "clclear"
jump menu
when "F"
execute "cpnclear"
jump menu
when "F"
execute "fscclear"
jump menu
when "G"
execute "ftsclear"
jump menu
when "H"
execute "invclear"
jump menu
when "I"
execute "othclear"
jump menu
when "J"
execute "sbclear"
jump menu
when "X"
  transfer "menu"
otherwise
  stop
end case
end main
CBCLEAR.PF3

global $gmatch $fmatch $respa $respb
public $gcode
main
repaint off
screen clear box 1 1 21 80 3 7
screen clear box 7 10 13 70 3 0
screen print 10 15 15 0 "Do you want to clear GL file?"
screen input 10 46 15 0 1 $respa
$gcode = "CB"
if $respa == "y"
  file load standard-view "cbgc.vws"
  order sort execute "debcrc.dfs" index "debcrc.idx"
  window split vertical 42
  data goto window 2
  file load standard-view "glclear.vws"
  data utilities append "cbgc.vws"
  keys F10
  screen clear 3 0
  repaint on

  data enter
  repaint off
  screen clear box 1 1 21 80 3 7
  screen clear box 7 10 13 70 3 0
  screen print 10 20 15 0 "Please wait . . . ."
  data transact execute "cbgcrlr.dft"
  data query execute "delall.dfq"
  REPLACE DELETE
  keys Esc
  window close
  file unload view "glclear.vws"
  data utilities purge "glclear.db"
  data goto view "cbgc.vws"
  order key rebuild
  data query execute "match.dfg" index "match.idx"
  NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
  keys Esc
  if cerror <> 3143
    $gmatch = filesum([sign])
    order change physical
  else
    $gmatch = 0
    order change physical
  end if
else
  file load standard-view "cbgc.vws"
  data query execute "match.dfg" index "match.idx"
  NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
  keys Esc
  if cerror <> 3143
    $gmatch = filesum([sign])
    order change physical
  else
    $gmatch = 0
  end if
screen clear box 1 1 21 80 3 7
screen clear box 7 10 13 70 3 0
screen print 10 15 15 0 "Do you want to clear FRB file?"
screen input 10 46 15 0 1 $respb
if $respb == "y"
  file load standard-view "cbfc.vws"
  order sort execute "debrcre.dfs" index "debcrc.idx"
  window split vertical 42
  data goto window 2
  file load standard-view "frbclear.vws"
  data utilities append "cbfc.vws"
  keys F10
  screen clear 3 0
  repaint on
  data enter
  repaint off
screen clear box 1 1 21 80 3 7
screen clear box 7 10 13 70 3 0
screen print 10 20 15 0 "Please wait . . . ."
data transact execute "cbfclr.dft"
data query execute "delall.dfq"'
REPLACE DELETE
keys Esc
window close
file unload view "frbclear.vws"
data utilities purge "frbclear.db"
data goto view "cbfc.vws"
order key rebuild
data query execute "match.dfq" index "match.idx"
',
NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
keys Esc
if error <> 3143
  $fmatch = filesum([sign])
  order change physical
else
  $fmatch = 0
end if
else
if $respa == "N"
jump end
else
  file load standard-view "cbfc.vws"
data query execute "match.dfq" index "match.idx"
',
NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
keys Esc
if error <> 3143
  $fmatch = filesum([sign])
  order change physical
else
  $fmatch = 0
end if
end if
label out
if round($gmatch + $fmatch,2) = 0
  screen clear box 1 1 21 80 3 7
screen clear box 7 10 13 70 3 0
screen print 10 20 15 0 " Cleared items net to zero"
wait 2
screen print 12 20 15 0 " Printed items will be deleted ".
window split vertical 42
data goto window 2
file load standard-view "clear.vws"
data goto window 1
data goto view "cbgc.vws"
data query execute "match.dfq" index "match.idx"
' NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
keys Esc
if error <> 3143
  data goto window 2
data utilities append "cbgc.vws"
keys Down, Down, Down, Down, Down, Down, F7, Down, Down, Right
keys Enter, F7, Down, Right, Right, Enter, Down, F10
data goto window 1
data query execute "matchdel.dfq"
  [MATCH] <> NULL REPLACE DELETE
keys Esc
file unload view "cbgc.vws"
data utilities purge "cbgc.db"
else
  file unload view "cbgc.vws"
end if

data goto view "cbfc.vws"
data query execute "match.dfq" index "match.idx"
' NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
keys Esc
if error <> 3143
  data goto window 2
data utilities append "cbfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right
keys Right, Right, Enter, Down, F7, Down, Right, Right, Enter, F10
data goto window 1
data query execute "matchdel.dfq"
  [MATCH] <> NULL REPLACE DELETE
keys Esc
file unload view "cbfc.vws"
data utilities purge "cbfc.db"
else
  file unload view "cbfc.vws"
end if

window close
data goto view "clear.vws"
order change key "[AMOUNT]"
print report execute "clear.dfr" printer detail start 1 end 0 copies

1
data query execute "delall.dfq"
REPLACE DELETE
keys Esc
file unload all
data utilities purge "clear.db"
else
screen clear box 1 1 21 80 3 7
screen clear box 7 10 13 70 3 0
screen print 10 20 15 0 " Cleared items do not net to zero"
wait 2
screen print 12 20 15 0 " Items will not be deleted from file"
window split vertical 42
data goto window 2
file load standard-view "clear.vws"
data goto window 1
data goto view "cbgc.vws"
data query execute "match.dfq" index "match.idx"
' NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
keys Esc
if cerror <> 3143
  data goto window 2
data utilities append "cbgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right
keys Enter, F7, Down, Right, Right, Right, Enter, Down, F10
  data goto window 1
data unload view "cbgc.vws"
else
  file unload view "cbgc.vws"
end if
data goto view "cbfc.vws"
data query execute "match.dfq" index "match.idx"
' NOT(ISBLANK([MATCH])) AND [MATCH] <> NULL
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right
keys Right, Right, Right, Enter, Down, F7, Down, Right, Right, Right, Enter, F10
data goto window 1
file unload view "cbfc.vws"
else
  file unload view "cbfc.vws"
end if
window close
data goto view "clear.vws"
order change key "[AMOUNT]"
print report execute "noclear.dfr" printer detail start 1 end 0
copies 1
data query execute "delall.dfq"
' REPLACE DELETE
keys Esc
if cerror <> 3958
file unload all
data utilities purge "clear.db"
else
file unload all
end if
end if
label end
data unload all
end main
RPTMENU.PF3

global $choice
main
file unload all
label menu
repaint on
screen clear box 1 1 21 80 0 5
screen clear box 2 14 20 67 0 0
screen draw box 2 14 20 67 15 0
screen print 3 25 11 0 "GL Account 1060 Reconciliation"
screen print 4 27 11 0 "Automated Matching Program"
screen print 5 27 11 0 "Reports Menu"
screen print 6 27 11 0 "="
screen print 7 19 3 0 "A) G/L Balancing Group file for date range"
screen print 8 19 3 0 "B) G/L Detailed Group file for date range"
screen print 9 19 3 0 "C) FRB Balancing Group file for date range"
screen print 10 19 3 0 "D) FRB Detailed Group file for date range"
screen print 11 19 3 0 "E) GL File replacements"
screen print 12 19 3 0 "F) FRB File replacements"
screen print 13 19 3 0 "G) All outstanding items"
screen print 14 19 3 0 "H) Outstanding items by detailed group"
screen print 15 19 3 0 "I) Stale Items by Detailed Group"
screen print 16 19 3 0 "J) Stale Items for All Groups"
screen print 17 19 3 0 "K) Cover Sheet"
screen print 18 19 3 0 "X) Return to main menu"
screen print 19 19 3 0 "Choice:"
screen input 19 27 3 0 1 $choice mask "[A-KXa-kx]"
case upper($choice)
  when "A"
    execute "glnrpt"
    jump menu
  when "B"
    execute "gldrpt"
    jump menu
  when "C"
    execute "frbrpt"
    jump menu
  when "D"
    execute "frbrpt"
    jump menu
  when "E"
    execute "glmod"
    jump menu
  when "F"
    execute "frbmod"
    jump menu
  when "G"
    execute "mergeo"
    jump menu
  when "H"
    execute "drpt"
    jump menu
  when "I"
    execute "stalebyg"
    jump menu
when "J"
  execute "stale"
  jump menu
when "K"
  execute "cover"
  jump menu
when "X"
  transfer "menu"
otherwise
  stop
end case
end main
GLRPT.PF3

public $gcode $date $date $drange

main
  repaint off
  screen clear box 1 1 21 80 0 5
  screen clear box 5 14 13 67 0 0
  screen draw box 5 14 14 67 15 0
  screen print 7 20 15 0 "Enter group code: 
  screen input 7 39 15 0 5 $gcode
  screen print 9 20 15 0 "Enter As-Of Date: 
  screen input 9 39 15 0 10 $date
  $date = date1($date)
  screen print 11 20 15 0 "Enter date range: 
  screen input 11 40 15 0 25 $drange

  if $gcode == "CB"
    file load standard-view "cbgc.vws"
  elseif $gcode == "ACH"
    file load standard-view "achgc.vws"
  elseif $gcode == "CC"
    file load standard-view "ccgc.vws"
  elseif $gcode == "CL"
    file load standard-view "clgc.vws"
  elseif $gcode == "CPN"
    file load standard-view "cpngc.vws"
  elseif $gcode == "FS"
    file load standard-view "fsgc.vws"
  elseif $gcode == "FTS"
    file load standard-view "ftsgc.vws"
  elseif $gcode == "INV"
    file load standard-view "invgc.vws"
  elseif $gcode == "OTHER"
    file load standard-view "othergc.vws"
  elseif $gcode == "SB"
    file load standard-view "sbgc.vws"
  else
    screen print 13 20 15 0 "Invalid group code . . ."
    wait 2.5
    repaint on
    transfer "rptmenu"
  end if
repaint on
data query modify "drange"

'DATE1([EFF_DATE])' >= date
AND
'DATE1([EFF_DATE])' <= date

repaint off
screen clear box 1 1 21 80 0 5
screen clear box 7 14 13 67 0 0
screen clear box 7 14 13 67 15 0
screen print 10 20 15 0 "Preparing report, please wait . . . ."
data query execute "drange" index "drange"
keys Enter,"drange",Enter,Esc

if error <> 3143
   order sort execute "words" index "words"
   print report execute "glfile.dfr" printer detail start 1 end 0 copies 1
   file unload all
else
   screen print 13 20 15 0 "No records found for date range selected."
   wait 1.5
   file unload all
end if
end main
GLDRPT.PF3

public $gcode $idate $date $drange

main
repaint off
screen clear box 1 1 21 80 0 5
screen clear box 5 14 14 67 0 0
screen draw box 5 14 14 67 15 0
screen print 7 20 15 0 "Enter detailed group code:"
screen input 7 48 15 0 5 $gcode
screen print 9 20 15 0 "Enter As-Of Date:"
screen input 9 39 15 0 10 $idate
$date = datel($idate)
screen print 11 20 15 0 "Enter date range:"
screen input 11 40 15 0 25 $drange

if ($gcode == "CBACH" or $gcode == "CBCL" or $gcode == "CBOTH")
  file load standard-view "cbgc.vws"
elseif ($gcode == "WACH" or $gcode == "EACH")
  file load standard-view "achgc.vws"
elseif ($gcode == "CCGH" or $gcode == "CCNA" or $gcode == "CCNV"
  or $gcode == "CCR" or $gcode == "CCSV")
  file load standard-view "ccgc.vws"
elseif ($gcode == "ECL" or $gcode == "WCL" or 
  $gcode == "ERET" or $gcode == "WRET")
  file load standard-view "clgc.vws"
elseif ($gcode == "WVCPN" or $gcode == "TRCPN")
  file load standard-view "cpngc.vws"
elseif ($gcode == "EVOC" or $gcode == "WVOC")
  file load standard-view "fsgc.vws"
elseif $gcode == "FTS"
  file load standard-view "ftsgc.vws"
elseif ($gcode == "MM" or $gcode == "AIM" or $gcode == "TRUST"
  or $gcode == "MBSPD" or $gcode == "MS" or $gcode == "INV")
  file load standard-view "invgc.vws"
elseif $gcode == "OTHER"
  file load standard-view "othergc.vws"
elseif ($gcode == "SBGH" or $gcode == "SBNA" or $gcode == "SBNV" or 
  $gcode == "SBR" or $gcode == "SBSV")
  file load standard-view "sbgc.vws"
else
  screen print 10 20 15 0 "Invalid group code . . ."
  wait 2.5
  repaint on
  transfer "rptmenu"
end if

repaint on
data query modify "cggroup"

'\[RGROUP\] = "group"
AND
'DATE1([EFF_DATE]) >= date
AND
'DATE1([EFF_DATE]) <= date

repaint off
screen clear box 1 1 21 80 0 5
screen clear box 7 14 13 67 0 0
screen draw box 7 14 13 67 15 0
screen print 10 20 15 0 "Preparing report, please wait . . . ."
data query execute "cggroup" index "cggroup"
keys Enter,"cggroup",Enter,Esc

if error <> 3143
  order sort execute "words" index "words"
  print report execute "glfile.dfr" printer detail start 1 end 0 copies 1
  file unload all
else
  screen print 13 16 15 0 "No records found for group and date range selected."
  wait 1.5
  file unload all
end if
end main
FRBRPT.PF3

public $gcode $date $date $drange

main
repaint off
screen clear box 1 1 21 80 0 5
screen clear box 5 14 11 67 0 0
screen draw box 5 14 14 67 15 0
screen print 7 20 15 0 "Enter group code: 
screen input 7 39 15 0 5 $gcode
screen print 9 20 15 0 "Enter As-Of Date: 
screen input 9 39 15 0 10 $date
$date = date($date)
screen print 11 20 15 0 "Enter date range: 
screen input 11 40 15 0 25 $drange

if $gcode == "CB"
  file load standard-view "cbfc.vws"
elseif $gcode =="ACH"
  file load standard-view "achfc.vws"
elseif $gcode =="CC"
  file load standard-view "ccfc.vws"
elseif $gcode =="CL"
  file load standard-view "clfc.vws"
elseif $gcode =="CPN"
  file load standard-view "cpnfc.vws"
elseif $gcode =="FS"
  file load standard-view "fsfc.vws"
elseif $gcode =="FTS"
  file load standard-view "ftsfc.vws"
elseif $gcode =="INV"
  file load standard-view "invfc.vws"
elseif $gcode =="OTHER"
  file load standard-view "otherfc.vws"
elseif $gcode =="SB"
  file load standard-view "sbfc.vws"
else
  screen print 13 20 15 0 "Invalid group code . . ."
  wait 2.5
  repaint on
  transfer "rptmenu"
end if

repaint on
data query modify "dfrange"
'DATE1([PRO_DATE]) >= date
'AND
'DATE1([PRO_DATE]) <= date

repaInt off
screen clear box 1 1 21 80 0 5
screen clear box 7 14 13 67 0 0
screen draw box 7 14 14 57 15 0
screen print 0 20 5 0 "Preparing report, please wait . . . ."
data query execute "dfrange" index "dfrange"
keys Enter,"dfrange",Enter,Esc

if cerror <> 3143
  order sort execute "wordx" index "wordx"
  print report execute "frbfie.defr" printer detail start 1 end 0 copies 1
else
  file unload all
end if
end main
FRDBRPT.PF3

public $gcode $date $date $drange

main
repaint off
screen clear box 1 1 21 80 0 5
screen clear box 5 14 14 67 0 0
screen draw box 5 14 14 67 15 0
screen print 7 20 15 0 "Enter detailed group code: 	" 
screen input 7 47 15 0 5 $gcode
screen print 9 20 15 0 "Enter As-Of Date: 	"
screen input 9 39 15 0 10 $date
$date = date:$($date)
screen print 11 20 15 0 "Enter date range: 	"
screen input 11 40 15 0 25 $drange

if ($gcode == "CBACH" or $gcode == "CBCL" or $gcode == "CBOOTH")
  file load standard-view "cbfc.vws"
elseif ($gcode =="WACH" or $gcode == "EACH")
  file load standard-view "achfc.vws"
elseif ($gcode == "CCGH" or $gcode == "CCNA" or $gcode == "CCNV"
  or $gcode == "CCR" or $gcode == "CCSV")
  file load standard-view "ccfc.vws"
elseif ($gcode == "ECL" or $gcode == "WCL" 
  or $gcode == "ETRT" or $gcode == "WRET")
  file load standard-view "clfc.vws"
elseif ($gcode == "WVCPN" or $gcode == "TRCPN")
  file load standard-view "cpnfc.vws"
elseif ($gcode == "EVOC" or $gcode == "WVOC")
  file load standard-view "fsfc.vws"
elseif $gcode == "FTS"
  file load standard-view "ftsf.c.vws"
elseif ($gcode == "MM" or $gcode == "AIM" or $gcode == "TRUST"
  or $gcode == "MBSPD" or $gcode == "MBS" or $gcode == "INV")
  file load standard-view "invfc.vws"
elseif $gcode == "OTHER"
  file load standard-view "otherfc.vws"
elseif ($gcode == "SBGH" or $gcode == "SBNA" or $gcode == "SBNV" or 
  $gcode == "SBR" or $gcode == "SBUV")
  file load standard-view "sbfc.vws"
else
  screen print 10 20 15 0 "Invalid group code . . .	"
  wait 2.5
  repaint on
  transfer "rptmenu"
end if

data query modify "cfgroup"

' [GROUP] = "group"
' AND
' DATE1([EFF_DATE]) >= date
' AND
' DATE1([EFF_DATE]) <= date

repaint off
screen clear box 1 1 21 80 0 5
screen clear box 7 14 13 67 0 0
screen draw box 7 14 13 67 15 0
screen print 10 20 15 0 "Preparing report, please wait ..."
data query execute "cfgroup" index "cfgroup"
keys Enter,"cfgroup",Enter,Esc

if error <> 3143
  order sort execute "wordx" index "wordx"
  print report execute "frbfile.dfr" printer detail start 1 end 0 copies 1
  file unload all
else
  screen print 13 16 15 0 "No records found for group and date range selected."
  wait 1.5
  file unload all
end if
end main
GLMOD.PP3

global $resp
main
repaing off
screen clear box 1 1 21 80 15 5
screen clear box 9 15 13 60 15 0
screen print 11 20 3 0 "Printing replacements report . . . ."

file load standard-view "modfileg"
order change key "[GROUP]"
print report execute "glmod" printer detail start 1 end 0 copies 1

screen clear box 9 15 13 60 15 0
screen print 11 20 3 0 "Do you want to empty file? "
screen input 11 47 3 0 1 $resp
if $resp == "Y"
data query execute "delall"
' REPLACE DELETE
keys Esc
file unload all
data utilities purge "modfileg.db"
else
file unload all
end if
repaing on
global $resp
main
repaint off
screen clear box 1 1 21 80 15 5
screen clear box 9 15 13 60 15 0
screen print 11 20 3 0 "Printing replacements report . . ."

file load standard-view "modfilef"
order change key "[GROUP]"
print report execute "frbmod" printer detail start 1 end 0 copies 1

screen clear box 9 15 13 60 15 0
screen print 11 20 3 0 "Do you want to empty file? "
screen input 11 47 3 0 1 $resp
if $resp == "Y"
data query execute "delall"
  'REPLACE DELETE
keys Esc
file unload all
data utilities purge "modfilef.db"
else
file unload all
end if
repaint on
MERGE.PF3

public $date $idate
main
screen clear box 1 1 21 80 0 5
screen clear box 7 14 11 67 0 0
screen draw box 7 14 11 67 15 0
screen print 9 25 3 0 "Input as-of date: 
screen input 9 43 3 0 10 $idate
$date = datel($idate)
repaaint off

file load standard-view "merge.vws"
data goto record last
if preorder <> 1
  data query execute "delall"
  keys Esc
  file unload all
  data utilities purge "merge.db"
  file load standard-view "merge.vws"
else
  jump cont
end if
label cont
window split vertical 42
file load standard-view "achgc.vws"
data goto window 2
data utilities append "achgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "achgc.vws"
file load standard-view "achfc.vws"
data goto window 2
data utilities append "achfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "achfc.vws"

file load standard-view "cbgc.vws"
data goto window 2
data utilities append "cbgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1

end if
file unload view "cbgc.vws"

file load standard-view "cbfc.vws"
data goto window 2
data utilities append "cbfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, P7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "cbfc.vws"

file load standard-view "ccgc.vws"
data goto window 2
data utilities append "ccgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, P7, Down, Down, Right, Enter
keys Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "ccgc.vws"
file load standard-view "ccfc.vws"
data goto window 2
data utilities append "ccfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "ccfc.vws"

file load standard-view "clgc.vws"
data goto window 2
data utilities append "clgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "clgc.vws"
file load standard-view "clfc.vws"
data goto window 2
data utilities append "clfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, P7, Down, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "clfc.vws"

file load standard-view "cpngc.vws"
data goto window 2
data utilities append "cpngc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "cpngc.vws"
file load standard-view "cnpfc.vws"
data goto window 2
data utilities append "cnpfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "cnpfc.vws"
file load standard-view "fsgc.vws"
data goto window 2
data utilities append "fsgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "fsgc.vws"
file load standard-view "fsfc.vws"
data goto window 2
data utilities append "fsfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
if cerror = 3958
data goto window 1
else
end if
file unload view "fsfc.vws"
file load standard-view "ftsfc.vws"
data goto window 2
data utilities append "ftsfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "ftsgc.vws"
file load standard-view "ftsfc.vws"
data goto window 2
data utilities append "ftsfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, P7, Down, Right, Right, Right
keys Right, Enter, Down, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "ftsfc.vws"
file load standard-view "invgc.vws"
data goto window 2
data utilities append "invgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, P7, Down, Down, Right, Enter
keys Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "invgc.vws"
file load standard-view "invfc.vws"
data goto window 2
data utilities append "invfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Right, Right
keys Right, Enter, Down, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "invfc.vws"
file load standard-view "othergc.vws"
data goto window 2
data utilities append "othergc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Enter
keys Down, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "othergc.vws"
file load standard-view "otherfc.vws"
data goto window 2
data utilities append "otherfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Right, Right
keys Right, Enter, Down, Down, Down, F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "otherfc.vws"

file load standard-view "sbgc.vws"
data goto window 2
data utilities append "sbgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "sbgc.vws"
file load standard-view "sbfc.vws"
data goto window 2
data utilities append "sbfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
if cerror = 3958
data goto window 1
else
data goto window 1
end if
file unload view "sbfc.vws"
data goto window 2
order sort execute "temp.dfs" index "temp.idx"
print report execute "outs.dfr" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
' data utilities purge "merge.db"
window close
repaint on
public $gcode $idate $date

main
repaint off
screen clear box 1 1 21 80 0 5
screen clear box 5 14 13 67 0 0
screen draw box 5 14 14 67 15 0
screen print 7 20 15 0 "Enter detailed group code: 
screen input 7 48 15 0 5 $gcode
screen print 9 20 15 0 "Enter As-Of Date: 
screen input 9 39 15 0 10 $idate
$date = date1($idate)

file load standard-view "merge.vws"
data goto record last
if precord <> 1
    data query execute "delall"
    keys Esc
    file unload all
    data utilities purge "merge.db"
else
    file unload all
    jump cont
end if
label cont

if ($gcode == "CBACH" or $gcode == "CBCL" or $gcode == "CBOTH")
    file load standard-view "cbgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "drpt.dfq" index "drpt.idx"
    keys Esc
    if error <> 3143
        data goto window 2
        data utilities append "cbgc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
    keys Down, Down, F10
    data goto window 1
    file unload view "cbgc.vws"
else
    data goto window 1
    file unload view "cbgc.vws"
end if

file load standard-view "cbfc.vws"
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if error <> 3143
    data goto window 2
    data utilities append "cbfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, Down, F10
    data goto window 1
    file unload view "cbfc.vws"
else
    data goto window 1
    file unload view "cbfc.vws"
end if

window close
order change key "]FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
    file unload all
else
    file unload all
end if

data utilities purge "merge.db"

elseif ($gcode == "WACH" or $gcode == "EACH")
    file load standard-view "achgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "drpt.dfq" index "drpt.idx"
    keys Esc
    if cerror <> 3143
        data goto window 2
        data utilities append "achgc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Enter
    keys Down, Down, F10
        data goto window 1
        file unload view "achgc.vws"
    else
        data goto window 1
        file unload view "cbgc.vws"
    end if

    file load standard-view "cbfc.vws"
    data query execute "drpt.dfq" index "drpt.idx"
    keys Esc
    if cerror <> 3143
        data goto window 2
        data utilities append "cbfc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Right, Right
    keys Right, Enter, Down, Down, F10
        data goto window 1
        file unload view "cbfc.vws"
    else
        data goto window 1
        file unload view "cbfc.vws"
    end if

window close
order change key "]FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
.data utilities purge "merge.db"
.elseif ($gcode == "CCGH" or $gcode == "CCNA" or $gcode == "CCNV" \ or $gcode == "CCR" or $gcode == "CCSV")
file load standard-view "ccgc.vws"
.window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ccgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "ccgc.vws"
else
data goto window 1
file unload view "ccgc.vws"
end if
file load standard-view "ccfc.vws"
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ccfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "ccfc.vws"
else
data goto window 1
file unload view "ccfc.vws"
end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
.data utilities purge "merge.db"
.elseif ($gcode =="ECL" or $gcode == "WCL" \ or $gcode == "ERET" or $gcode == "WRET")
file load standard-view "clgc.vws"
.window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "clgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "clgc.vws"
else
data goto window 1
file unload view "clgc.vws"
end if
file load standard-view "clfc.vws"
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "clfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "clfc.vws"
else
data goto window 1
file unload view "clfc.vws"
end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif ($code == "WVCPN" or $code == "TRCPN")
file load standard-view "cpnc.vws"
window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cpngc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
```plaintext
file unload view "cpncg.vws"
else
  data goto window 1
  file unload view "cpncg.vws"
end if

file load standard-view "cplnfc.vws"
data query execute "drpt.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cplnfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,Down,F10
data goto window 1
  file unload view "cplnfc.vws"
else
  data goto window 1
  file unload view "cplnfc.vws"
end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
  file unload all
end if
data utilities purge "merge.db"
elseif ($gcode == "EVOC" or $gcode == "WVOC")
  file load standard-view "fsgc.vws"
  window split vertical 42
  data goto window 2
  file load standard-view "merge.vws"
  data goto window 1
  data query execute "drpt.dfq" index "drpt.idx"
  keys Esc
  if cerror <> 3143
    data goto window 2
    data utilities append "fsgc.vws"
  keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Down,Down,Right,Enter
  keys Down,Down,F10
data goto window 1
  file unload view "fsgc.vws"
else
  data goto window 1
  file unload view "fsgc.vws"
end if

file load standard-view "fsfc.vws"
data query execute "drpt.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
data goto window 2
```
data utilities append "fsfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
data goto window 1
  file unload view "fsfc.vws"
else
data goto window 1
  file unload view "fsfc.vws"
end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif $gcode == "FTS"
  file load standard-view "ftsgc.vws"
  window split vertical 42
data goto window 2
  file load standard-view "merge.vws"
data goto window 1
data query execute "drpt.dfq" index "drpt.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ftsgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
data goto window 1
  file unload view "ftsgc.vws"
else
data goto window 1
  file unload view "ftsgc.vws"
end if

  file load standard-view "ftsf.vws"
data query execute "drpt.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ftsf.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
data goto window 1
  file unload view "ftsf.vws"
else
data goto window 1
  file unload view "ftsf.vws"
end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif ($scode = "MM" or $scode = "AIM" or $scode ="TRUST" \ or $scode = "MBSPD" or $scode = "MBS" or $scode = "INV")
  file load standard-view "invgc.vws"
  window split vertical 42
  data goto window 2
  file load standard-view "merge.vws"
  data goto window 1
  data query execute "drpt.dfq" index "drpt.idx"
  keys Esc
  if cerror <> 3143
    data goto window 2
    data utilities append "invgc.vws"
  keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
  keys Down,Down,F10
    data goto window 1
    file unload view "invgc.vws"
  else
    data goto window 1
    file unload view "invgc.vws"
  end if

  file load standard-view "invfc.vws"
  data query execute "drpt.dfq" index "agefrb.idx"
  keys Esc
  if cerror <> 3143
    data goto window 2
    data utilities append "invfc.vws"
  keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
  keys Right,Enter,Down,Down,F10
    data goto window 1
    file unload view "invfc.vws"
  else
    data goto window 1
    file unload view "invfc.vws"
  end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"
elseif $gcode == "OTHER"
    file load standard-view "othergc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "drpt.dfq" index "drpt.idx"
    keys Esc
    if cerror <> 3143
        data goto window 2
        data utilities append "othergc.vws"
    end if
    keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
    keys Down,Down,F10
    data goto window 1
    file unload view "othergc.vws"
else
    data goto window 1
    file unload view "othergc.vws"
end if

file load standard-view "otherfc.vws"
data query execute "drpt.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
    data goto window 2
    data utilities append "otherfc.vws"
end if
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
    data goto window 1
    file unload view "otherfc.vws"
else
    data goto window 1
    file unload view "otherfc.vws"
end if
window close
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
    file unload all
else
    file unload all
end if
data utilities purge "merge.db"

elseif ($gcode == "SBGH" or $gcode == "SBNA" or $gcode == "SBNV" or \
        $gcode == "SBR" or $gcode == "SBSV")
    file load standard-view "sbgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "drpt.dfq" index "drpt.idx"
    keys Esc
    if cerror <> 3143
        data goto window 2
    else

data utilities append "sbgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
  file unload view "sbgc.vws"
else
data goto window 1
  file unload view "sbgc.vws"
end if

data load standard-view "sbfc.vws"
data query execute "drpt.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
data goto window 2
  data utilities append "sbfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,Down,Right,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
  file unload view "sbfc.vws"
else
data goto window 1
  file unload view "sbfc.vws"
end if
window close
order key rebuild
order change key "[FILE]"
print report execute "gmerge" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror <= 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

else
  screen print 10 20 15 0 "Invalid group code . . ."
  wait 2.5
  repaint on
  transfer "rptmenu"
end if
repaint on
end main
public $gcode $idate $date

main
  repaint off
  screen clear box 11 21 80 0 5
  screen clear box 5 14 13 67 0 0
  screen draw box 5 14 14 67 15 0
  screen print 7 20 15 0 "Enter detailed group code:"
  screen input 7 48 15 0 5 $gcode
  screen print 9 20 15 0 "Enter As-Of Date:"
  screen input 9 39 15 0 10 $idate
  $date = date1($idate)

  file load standard-view "merge.vws"
  data goto record last
  if precord <> 1
    data query execute "delall"
    keys Esc
    file unload all
    data utilities purge "merge.db"
  else
    file unload all
    jump cont
  end if
  label cont

  if ($gcode == "CBACH" or $gcode == "CBCL" or $gcode == "CBOTH")
    file load standard-view "cbgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "age7gl.dfq" index "agegl.idx"
    keys Esc
    if cerror <> 3143
      data goto window 2
      data utilities append "cbgc.vws"
      keys Down, Down, Down, Down, Down, Down, Down, P7, Down, Down, Right, Enter
      keys Down, Down, F10
      data goto window 1
      file unload view "cbgc.vws"
    else
      data goto window 1
      file unload view "cbgc.vws"
    end if

    file load standard-view "cbfc.vws"
    data query execute "age7frb.dfq" index "agefrb.idx"
    keys Esc
    if cerror <> 3143
      data goto window 2
      data utilities append "cbfc.vws"
      keys Down, Down, Down, Down, Down, Down, Down, Down, P7, Down, Right, Right, Right
      keys Right, Enter, Down, Down, F10
      data goto window 1
      file unload view "cbfc.vws"
else
    data goto window 1
    file unload view "cbfc.vws"
end if
window close
order change key "[RGROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif ($gcode == "WACH" or $gcode == "EACH")
    file load standard-view "achgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "age2gl.dfq" index "agegl.idx"
    keys Esc
    if cerror <> 3143
    data goto window 2
    data utilities append "achgc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
    keys Down, Down, F10
    data goto window 1
    file unload view "achgc.vws"
    else
    data goto window 1
    file unload view "cbgc.vws"
end if

    file load standard-view "cbfc.vws"
    data query execute "age2frb.dfq" index "agefrb.idx"
    keys Esc
    if cerror <> 3143
    data goto window 2
    data utilities append "cbfc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right, Right
    keys Right, Enter, Down, Down, Down, F10
    data goto window 1
    file unload view "cbfc.vws"
    else
    data goto window 1
    file unload view "cbfc.vws"
end if
window close
order change key "[RGROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif ($gcode == "CCGH" or $gcode == "CCNA" or $gcode == "CCNV" \
     or $gcode == "CCR" or $gcode == "CCSV")
file load standard-view "ccgc.vws"
window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "age7gl.dfq" index "agegl.idx"
keys Esc
if cerror <> 3143
  data goto window 2
  data utilities append "ccgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
  data goto window 1
  file unload view "ccgc.vws"
else
  data goto window 1
  file unload view "ccgc.vws"
end if

file load standard-view "ccfc.vws"
data query execute "age7frb.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
  data goto window 2
  data utilities append "ccfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
  data goto window 1
  file unload view "ccfc.vws"
else
  data goto window 1
  file unload view "ccfc.vws"
end if
window close
order change key "[RGROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif ($gcode =="BCL" or $gcode == "WCL" \
     or $gcode =="BBET" or $gcode == "WRET")
file load standard-view "clgc.vws"
window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "age14gl.dfq" index "agegl.idx"
keys Esc
if error <> 3143
data goto window 2
data utilities append "clgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "clgc.vws"
else
data goto window 1
file unload view "clgc.vws"
end if

file load standard-view "clfc.vws"
data query execute "age14frb.dfq" index "agefrb.idx"
keys Esc
if error <> 3143
data goto window 2
data utilities append "clfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "clfc.vws"
else
data goto window 1
file unload view "clfc.vws"
end if

window close
order change key "$[GROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "detail"
keys Esc
if error = 3143
file unload all
else
file unload all
end if

data utilities purge "merge.db"

elseif ($gcode == "WVCPN" or $gcode == "TRCPN")
file load standard-view "cpncg.vws"
window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "age7gl.dfq" index "agegl.idx"
keys Esc
if error <> 3143
data goto window 2
data utilities append "cpncg.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "cpngc.vws"
else
   data goto window 1
   file unload view "cpngc.vws"
end if

file load standard-view "cpnfc.vws"
data query execute "age7frb.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
   data goto window 2
   data utilities append "cpnfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
   data goto window 1
   file unload view "cpnfc.vws"
else
   data goto window 1
   file unload view "cpnfc.vws"
end if
window close
order change key "[RGROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
   file unload all
else
   file unload all
end if
data utilities purge "merge.db"

elseif ($gcode == "EVOC" or $gcode == "WVOC")
   file load standard-view "fsgc.vws"
   window split vertical 42
   data goto window 2
   file load standard-view "merge.vws"
   data goto window 1
   data query execute "age14gl.dfq" index "agegl.idx"
   keys Esc
   if cerror <> 3143
      data goto window 2
      data utilities append "fsgc.vws"
   keys Down,Down,Down,Down,Down,Down,Down,Down,Down,Right,Enter
   keys Down,Down,F10
      data goto window 1
      file unload view "fsgc.vws"
   else
      data goto window 1
      file unload view "fsgc.vws"
   end if
   file load standard-view "fsfc.vws"
   data query execute "age14frb.dfq" index "agefrb.idx"
   keys Esc
   if cerror <> 3143
      data goto window 2

data utilities append "fsfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
    data goto window 1
    file unload view "fsfc.vws"
else
    data goto window 1
    file unload view "fsfc.vws"
end if
window close
order change key "{RGROUP}"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
    file unload all
else
    file unload all
end if
data utilities purge "merge.db"

elseif $gcode == "FTS"
    file load standard-view "ftsgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "age2gl.dfq" index "agegl.idx"
    keys Esc
    if cerror <= 3143
        data goto window 2
        data utilities append "ftsgc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Down, Right, Enter
    keys Down, Down, F10
        data goto window 1
        file unload view "ftsgc.vws"
    else
        data goto window 1
        file unload view "ftsgc.vws"
    end if

    file load standard-view "ftsfc.vws"
    data query execute "age2frb.dfq" index "agefrb.idx"
    keys Esc
    if cerror <= 3143
        data goto window 2
        data utilities append "ftsfc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
    keys Right, Enter, Down, Down, Down, F10
        data goto window 1
        file unload view "ftsfc.vws"
    else
        data goto window 1
        file unload view "ftsfc.vws"
    end if
window close
order change key "{RGROUP}"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"

elseif ($gcode == "MM" or $gcode == "AIM" or $gcode == "TRUST" \ or $gcode == "MBSPD" or $gcode == "MBS" or $gcode == "INV")
file load standard-view "invgc.vws"
window split vertical 42
data goto window 2
file load standard-view "merge.vws"
data goto window 1
data query execute "age14gl.dfq" index "agegl.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "invgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
data goto window 1
file unload view "invgc.vws"
else
data goto window 1
file unload view "invgc.vws"
end if

file load standard-view "invcf.vws"
data query execute "age14frb.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "invcf.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, Right, Right, Right
keys Right, Enter, Down, Down, Down, Down, F10
data goto window 1
file unload view "invcf.vws"
else
data goto window 1
file unload view "invcf.vws"
end if
window close
order change key "[RGROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"
elseif $gcode == "OTHER"
    file load standard-view "othergc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "age7gl.dfq" index "agegl.idx"
    keys Esc
    if error <> 3143
        data goto window 2
        data utilities append "othergc.vws"
    keys Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
    keys Down, Down, F10
        data goto window 1
        file unload view "othergc.vws"
    else
        data goto window 1
        file unload view "othergc.vws"
    end if

    file load standard-view "otherfc.vws"
    data query execute "age7frb.dfq" index "agefrb.idx"
    keys Esc
    if error <> 3143
        data goto window 2
        data utilities append "otherfc.vws"
    keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
    keys Right, Enter, Down, Down, F10
        data goto window 1
        file unload view "otherfc.vws"
    else
        data goto window 1
        file unload view "otherfc.vws"
    end if

    window close
    order change key "[RGROUP]"
    print report execute "stale" printer detail start 1 end 0 copies 1
    data query execute "delall"
    keys Esc
    if error = 3143
        file unload all
    else
        file unload all
    end if
    data utilities purge "merge.db"

elseif ($gcode == "SBGH" or $gcode == "SBNA" or $gcode == "SBNV" or \ $gcode == "SBR" or $gcode == "SBSV")
    file load standard-view "sbgc.vws"
    window split vertical 42
    data goto window 2
    file load standard-view "merge.vws"
    data goto window 1
    data query execute "age7gl.dfq" index "agegl.idx"
    keys Esc
    if error <> 3143
        data goto window 2
data utilities append "sbgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
    data goto window 1
    file unload view "sbgc.vws"
else
    data goto window 1
    file unload view "sbgc.vws"
end if

file load standard-view "sbfc.vws"
data query execute "age7frb.dfq" index "agefrb.idx"
keys Esc
if cerror <> 3143
    data goto window 2
    data utilities append "sbfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,Down,Right,Right,Right
keys Right,Enter,Down,Down,Down,F10
    data goto window 1
    file unload view "sbfc.vws"
else
    data goto window 1
    file unload view "sbfc.vws"
end if
window close
order change key "[RGROUP]"
print report execute "stale" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"
else
    screen print 10 20 15 0 "Invalid group code . . . "
    wait 2.5
    repaint on
    transfer "rptmenu"
end if
repaint on
end main
public $date $idate
main
quiet off
screen clear box 1 1 21 80 0 5
screen clear box 7 14 11 67 0 0
screen draw box 7 14 11 67 15 0
screen print 9 25 3 0 "Enter As-Of Date:"
screen input 9 43 3 0 10 $idate
$idate = datel($idate)
repaint off

file load standard-view "merge.vws"
data goto record last
if precord <> 1
    data query execute "delall"
    keys Esc
    file unload all
    data utilities purge "merge.db"
    file load standard-view "merge.vws"
else
    jump cont
end if
label cont
window split vertical 42
file load standard-view "achgc.vws"
data query execute "age2g" index "ageg1"
'DAYS($date) - DAYS(EFF_DATE) >= 2
keys Esc
if cerror <> 3143
    data goto window 2
data utilities append "achgc.vws"
    keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
    keys Down,Down,F10
data goto window 1
    file unload view "achgc.vws"
else
    file unload view "achgc.vws"
end if

file load standard-view "achfc.vws"
data query execute "age2f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 2
keys Esc
if cerror <> 3143
    data goto window 2
data utilities append "achfc.vws"
    keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
    keys Right,Enter,Down,Down,F10
    file unload view "achfc.vws"
data goto window 1
else
    file unload view "achfc.vws"
end if

data goto window 1
file load standard-view "cbgc.vws"
data query execute "age7g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cbgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "cbgc.vws"
else
file unload view "cbgc.vws"
end if

file load standard-view "cbfc.vws"
data query execute "age7f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cbfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "cbfc.vws"
else
file unload view "cbfc.vws"
end if

data goto window 1
file load standard-view "ccgc.vws"
data query execute "age7g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ccgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "ccgc.vws"
else
file unload view "ccgc.vws"
end if

file load standard-view "ccfc.vws"
data query execute "age7f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ccfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "ccfc.vws"
else
file unload view "ccfc.vws"
end if

data goto window 1
file load standard-view "clgc.vws"
data query execute "age14g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 14
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "clgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "clgc.vws"
else
file unload view "clgc.vws"
end if

file load standard-view "clfc.vws"
data query execute "age14f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 14
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "clfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "clfc.vws"
else
file unload view "clfc.vws"
end if

data goto window 1
file load standard-view "cpngc.vws"
data query execute "age7g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cpngc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
data goto window 1
file unload view "cpngc.vws"
else
file unload view "cpngc.vws"
end if

file load standard-view "cpnfc.vws"
data query execute "age7f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "cpnfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
data goto window 1
file unload view "cpnfc.vws"
else
file unload view "cpnfc.vws"
end if

data goto window 1
file load standard-view "fsgc.vws"
data query execute "age2g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 2
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "fsgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
data goto window 1
file unload view "fsgc.vws"
else
file unload view "fsgc.vws"
end if

file load standard-view "fsfc.vws"
data query execute "age2f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 2
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "fsfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
data goto window 1
file unload view "fsfc.vws"
else
file unload view "fsfc.vws"
data goto window 1
file load standard-view "ftsgc.vws"
data query execute "age2g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 2
if cerror <> 3143
data goto window 2
data utilities append "ftsgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "ftsgc.vws"
else
file unload view "ftsgc.vws"
end if

file load standard-view "ftsfc.vws"
data query execute "age2f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 2
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "ftsfc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Right,Right,Right
keys Right,Enter,Down,Down,F10
data goto window 1
file unload view "ftsfc.vws"
else
file unload view "ftsfc.vws"
end if

data goto window 1
file load standard-view "invgc.vws"
data query execute "age14g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 14
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "invgc.vws"
keys Down,Down,Down,Down,Down,Down,Down,Down,F7,Down,Down,Right,Enter
keys Down,Down,F10
data goto window 1
file unload view "invgc.vws"
else
file unload view "invgc.vws"
end if

file load standard-view "invfc.vws"
data query execute "age14f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 14
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "invfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
data goto window 1
file unload view "invfc.vws"
else
file unload view "invfc.vws"
end if

data goto window 1
file load standard-view "othergc.vws"
data query execute "age7g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "othergc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
data goto window 1
file unload view "othergc.vws"
else
file unload view "othergc.vws"
end if

file load standard-view "otherfc.vws"
data query execute "age7f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "otherfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, F7, Down, Right, Right, Right
keys Right, Enter, Down, Down, F10
data goto window 1
file unload view "otherfc.vws"
else
file unload view "otherfc.vws"
end if

data goto window 1
file load standard-view "sbgc.vws"
data query execute "age7g" index "agegl"
'DAYS($date) - DAYS(EFF_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "sbgc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Enter
keys Down, Down, F10
data goto window 1
file unload view "sbgc.vws"
else
file unload view "sbgc.vws"
end if

file load standard-view "sbfc.vws"
data query execute "age7f" index "agefrb"
'DAYS($date) - DAYS(PRO_DATE) >= 7
keys Esc
if cerror <> 3143
data goto window 2
data utilities append "sbfc.vws"
keys Down, Down, Down, Down, Down, Down, Down, Down, Down, F7, Down, Down, Right, Right, Right
keys Right, Enter, Down, Down, Down, F10
data goto window 1
file unload view "sbfc.vws"
else
file unload view "sbfc.vws"
end if

data goto window 2
order sort execute "temp.dfs" index "temp.idx"
print report execute "staleg" printer detail start 1 end 0 copies 1
data query execute "delall"
keys Esc
if cerror = 3143
file unload all
else
file unload all
end if
data utilities purge "merge.db"
repaent on
global $glcb $glach $glcc $glcl $glcpn $glfs $glfts $glinv $glother $glsb
global $frbcb $frbach $frbcc $frbcl $frbcpn $frbfs $frbfits $frbinv
global $frbother $frbsb
global $date $date $gl $frb $gllbal $frbbal

main
screen clear box 1 1 21 80 15 3
screen clear box 4 10 11 70 5 0
screen print 7 25 15 0 "Calculating . . ."
repaint off

file load standard-view "cbgc.vws"
$glcb = round(filesum([sign]),2)
file unload all

file load standard-view "cbfc.vws"
$frbcb = round(filesum([sign]),2)
file unload all

file load standard-view "achgc.vws"
$glach = round(filesum([sign]),2)
file unload all

file load standard-view "achfc.vws"
$frbach = round(filesum([sign]),2)
file unload all

file load standard-view "ccgc.vws"
$glcc = round(filesum([sign]),2)
file unload all

file load standard-view "ccfc.vws"
$frbcc = round(filesum([sign]),2)
file unload all

file load standard-view "clgc.vws"
$glcl = round(filesum([sign]),2)
file unload all

file load standard-view "-clfc.vws"
$frbcl = round(filesum([sign]),2)
file unload all
file load standard-view "cpncg.c.vws"
$glcpn = round(filesum([sign]),2)
file unload all

file load standard-view "cpnfc.vws"
$frbcpgn = round(filesum([sign]),2)
file unload all

file load standard-view "fsgc.vws"
$glfss = round(filesum([sign]),2)
file unload all

file load standard-view "fsfc.vws"
$frbfss = round(filesum([sign]),2)
file unload all

file load standard-view "ftsgc.vws"
$glftss = round(filesum([sign]),2)
file unload all

file load standard-view "ftsf.vws"
$frbftss = round(filesum([sign]),2)
file unload all

file load standard-view "invgc.vws"
$glinv = round(filesum([sign]),2)
file unload all

file load standard-view "invfc.vws"
$frbinv = round(filesum([sign]),2)
file unload all

file load standard-view "othergc.vws"
$glother = round(filesum([sign]),2)
file unload all

file load standard-view "otherfc.vws"
$frbother = round(filesum([sign]),2)
file unload all

file load standard-view "sbgc.vws"
$glsb = round(filesum([sign]),2)
file unload all

file load standard-view "sbfv.vws"
$frbsb = round(filesum([sign]),2)
  file unload all

screen clear box 4 10 11 70 15 0
screen print 5 20 15 0 "Enter As-Of Date: "
screen input 5 39 7 0 10 $date
$date = date($date)

screen print 7 20 15 0 "Enter GL Balance: "
screen input 7 39 7 0 14 $gl
$glbal = round(value($gl),2)

screen print 9 20 15 0 "Enter Fed Balance: "
screen input 9 39 7 0 14 $frb
$frbbal = round(value($frb),2)

file load standard-view "talleyc.vws"
data goto record first
[DATE] = $date

[GLOBAL] = $glbal
[FRBBAL] = $frbbal

[GLCB] = $glcb
[FRBCB] = $frbcb

[GLACH] = $glach
[FRBACH] = $frbach

[GLCC] = $glcc
[FRBCC] = $frbccc

[GLCL] = $glcl
[FRBCL] = $frbcl

[GLCPN] = $glcpn
[FRBCPN] = $frbcpn

[GLFS] = $glfs
[FRBFS] = $frbfs

[GLFTS] = $glfts
[FRBFTS] = $frbfts

[GLINV] = $glinv
[FRBINV] = $frbinv

[GLOTHER] = $glother
[FRBOTHER] = $frbother

[GLSB] = $glsb
[FRBSB] = $frbsb
print report execute "cover.dfr" printer detail start 1 end 0 copies 1 file unload all
RESTORE.PF3

global $resp $respa $respb $respc
main
'Clear screen
repaint off
screen clear 1 0
screen draw box 10 10 14 70 5 0
screen print 12 15 15 0 "Are you sure you want to restore files?"
screen input 12 56 15 0 1 $resp
if $resp == "Y"

    screen print 12 15 15 0 "Restore from hard drive (H) or diskettes (D)?"
    screen input 12 65 15 0 1 $respa

    if $respa == "H"
        tools file copy "c:\auto\restore\cbgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\cbfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\achgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\achfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\ccgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\ccfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\clgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\clfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\cpngc.*" to "c:\auto"
        tools file copy "c:\auto\restore\cpnfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\fsgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\fsfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\ftsgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\ftsfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\invgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\invfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\othergc.*" to "c:\auto"
        tools file copy "c:\auto\restore\otherfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\sbgc.*" to "c:\auto"
        tools file copy "c:\auto\restore\sbfc.*" to "c:\auto"
        tools file copy "c:\auto\restore\talleyg.*" to "c:\auto"
        tools file copy "c:\auto\restore\talleyf.*" to "c:\auto"
    elseif $respa == "D"
    screen print 12 15 15 0 "Insert Diskette 1, enter 'Y' when ready"
    screen input 12 56 15 0 1 $respb
    if $respb == "Y"
tools file copy "a:\cbgc.*" to "c:\auto"
tools file copy "a:\cbfc.*" to "c:\auto"

tools file copy "a:\achgc.*" to "c:\auto"
tools file copy "a:\achfc.*" to "c:\auto"

tools file copy "a:\ccgc.*" to "c:\auto"
tools file copy "a:\ccfc.*" to "c:\auto"

tools file copy "a:\clgc.*" to "c:\auto"
tools file copy "a:\clfc.*" to "c:\auto"

tools file copy "a:\cpngc.*" to "c:\auto"
tools file copy "a:\cpnf.*" to "c:\auto"

screen print 12 15 15 0 "Insert Diskette 2, enter 'Y' when ready"
screen input 12 56 15 0 1 $respc

if $respc == "Y"

 tools file copy "a:\fsgc.*" to "c:\auto"
tools file copy "a:\fsfc.*" to "c:\auto"

 tools file copy "a:\ftsngc.*" to "c:\auto"
tools file copy "a:\ftsf.*" to "c:\auto"

 tools file copy "a:\invgc.*" to "c:\auto"
tools file copy "a:\invfc.*" to "c:\auto"

 tools file copy "a:\othergc.*" to "c:\auto"
tools file copy "a:\otherfc.*" to "c:\auto"

 tools file copy "a:\sbgc.*" to "c:\auto"
tools file copy "a:\sbf.*" to "c:\auto"

 tools file copy "a:\talleyg.*" to "c:\auto"
tools file copy "a:\tallyf.*" to "c:\auto"

else
jump end
end if

else
jump end
end if

else
jump end
end if

else
jump end
end if
global $resp
main
  'Clear screen
  repaint off

  screen clear 1 0
  screen draw box 10 10 14 70 5 0
  screen print 12 15 15 0 "Insert Diskette 1, enter 'Y' when ready"
  screen input 12 56 15 0 1 $resp

  'Copy ten files to diskette 1
  tools file copy "c:\auto\cbgc.*" to "a:\"
  tools file copy "c:\auto\cbfc.*" to "a:\"
  tools file copy "c:\auto\achgc.*" to "a:\"
  tools file copy "c:\auto\achfc.*" to "a:\"
  tools file copy "c:\auto\ccgc.*" to "a:\"
  tools file copy "c:\auto\ccfc.*" to "a:\"
  tools file copy "c:\auto\clgc.*" to "a:\"
  tools file copy "c:\auto\clfc.*" to "a:\"
  tools file copy "c:\auto\cpnqc.*" to "a:\"
  tools file copy "c:\auto\cpnfc.*" to "a:\"

  screen clear 1 0
  sound 1000
  sound 2000
  sound 3000
  screen draw box 10 10 14 70 5 0
  screen print 12 15 15 0 "Insert Diskette 2, enter 'Y' when ready"
  screen input 12 56 15 0 1 $resp

  tools file copy "c:\auto\fsqc.*" to "a:\"
  tools file copy "c:\auto\fsfc.*" to "a:\"
  tools file copy "c:\auto\ftsgc.*" to "a:\"
  tools file copy "c:\auto\ftsfc.*" to "a:\"
  tools file copy "c:\auto\invgc.*" to "a:\"
  tools file copy "c:\auto\invfc.*" to "a:\"
  tools file copy "c:\auto\othergc.*" to "a:\"
  tools file copy "c:\auto\otherfc.*" to "a:\"
  tools file copy "c:\auto\sbgc.*" to "a:\"
  tools file copy "c:\auto\sbfc.*" to "a:\"
  tools file copy "c:\auto\talleyg.*" to "a:\"
  tools file copy "c:\auto\talleyf.*" to "a:\"
else
jump end
end if
label end
end main
ENDNOTES


2. Thomas M. Smith, p. 35.