The Development of the New Feeder System: Applying the System Development Life Cycle Methodology

Michael Menefield
Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

**DEPARTMENT OF COMPUTER SCIENCE
& SYSTEMS ANALYSIS**

TECHNICAL REPORT: MU-SEAS-CSA-1997-003

The Development of the New Feeder System: Applying the System Development Life Cycle Methodology

Michael Menefield

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

The Development of the New Feeder System: Applying the System Development Life Cycle Methodology

by

Michael Menefield
Systems Analysis Department
Miami University
Oxford, Ohio 45056

Working Paper #97-003 07/97

MIAMI UNIVERSITY

THE DEVELOPMENT OF
THE NEW FEEDER SYSTEM:
APPLYING THE
SYSTEM DEVELOPMENT LIFE CYCLE METHODOLOGY

A MASTER'S PROJECT SUBMITTED TO
THE FACULTY OF THE DIVISION OF APPLIED SCIENCE
IN CANDIDACY FOR THE DEGREE OF
MASTERS OF SYSTEM ANALYSIS

DEPARTMENT OF SYSTEM ANALYSIS

BY

MICHAEL C. MENEFIELD

OXFORD, OHIO
July, 1997

THE DEVELOPMENT OF THE NEW FEEDER SYSTEM: APPLYING THE SYSTEM DEVELOPMENT LIFE CYCLE METHODOLOGY

Table of Contents

Table of Contents (continued)

Abstract

My graduate project proposal is the development of the new Feeder system. The new Feeder system is comprised of five individual feeders, which are the Hourly Payroll, Salary Payroll, Stores, Job Labor and Accounts Payable feeders. My task is to effectively apply the methodology of the system development life cycle to this project. The purpose of this paper is to demonstrate my ability to identify and solve a real world problem using the theories and skills learned in the Master of System Analysis program. This paper begins by focusing on the problems that resulted in the development of a new Feeder system. It answers the question of "Why should a new system be developed?" Also, the paper outlines the methodology applied in the development of the new system. This addresses the question of "How did I develop the new system?" Next, I address the learning experiences in developing this project. In addition, I will outline the special features and the cost/benefit analysis of the new system.

Chapter 1

This chapter focuses on the problems that resulted in developing the new Feeder system. It will cover the following sections: introduction, financial system's problems, previous system overview and problems of the previous Feeder System. Also, it contains the previous systems: entity relationship diagrams, the decomposition diagrams and process model diagrams.

Introduction

The Middletown Works and Ashland Works cost accountancy departments of the AK Steel Corporation are responsible for gathering and computing the production cost of all the facilities and departments located at the respective sites. The primary functions of the cost accountancy departments are to gather its plant sites associated facilities cost and to produce weekly, monthly and annual cost center and departmental expenditure information.

Each of the cost accountancy departments has access to its own Operating Cost suprasystem and Feeder system to facilitate the task of gathering, computing and reporting expenditure information. The Operating Cost system is used to help derive the plant's expenditures on the AK Steel Corporation's Profit and Loss (P & L) statement. The Feeder system, a sub-system of the General Ledger suprasystem, is used to gather, edit and calculate data from several of the plant's financial systems. In addition, it is used to convert and construct acceptable journal vouchers, which are to be passed to the primary General Ledger system and the Operating Cost system, and to produce monthly expenditure information. (See Diagram A)

The previous Feeder system consisted of transaction files and a few PLI, SAS and CLIST batch programs. This sub-system was maintained on the IBM mainframe, which ran on a MVS operating system. Most of the code in this system had not been modified in over 10 years, which caused a great deal of effort in enhancing and maintaining this system.

The majority of AK Steel Corporation's newer systems were rewritten using DB2 tables and

High Level Data Flow

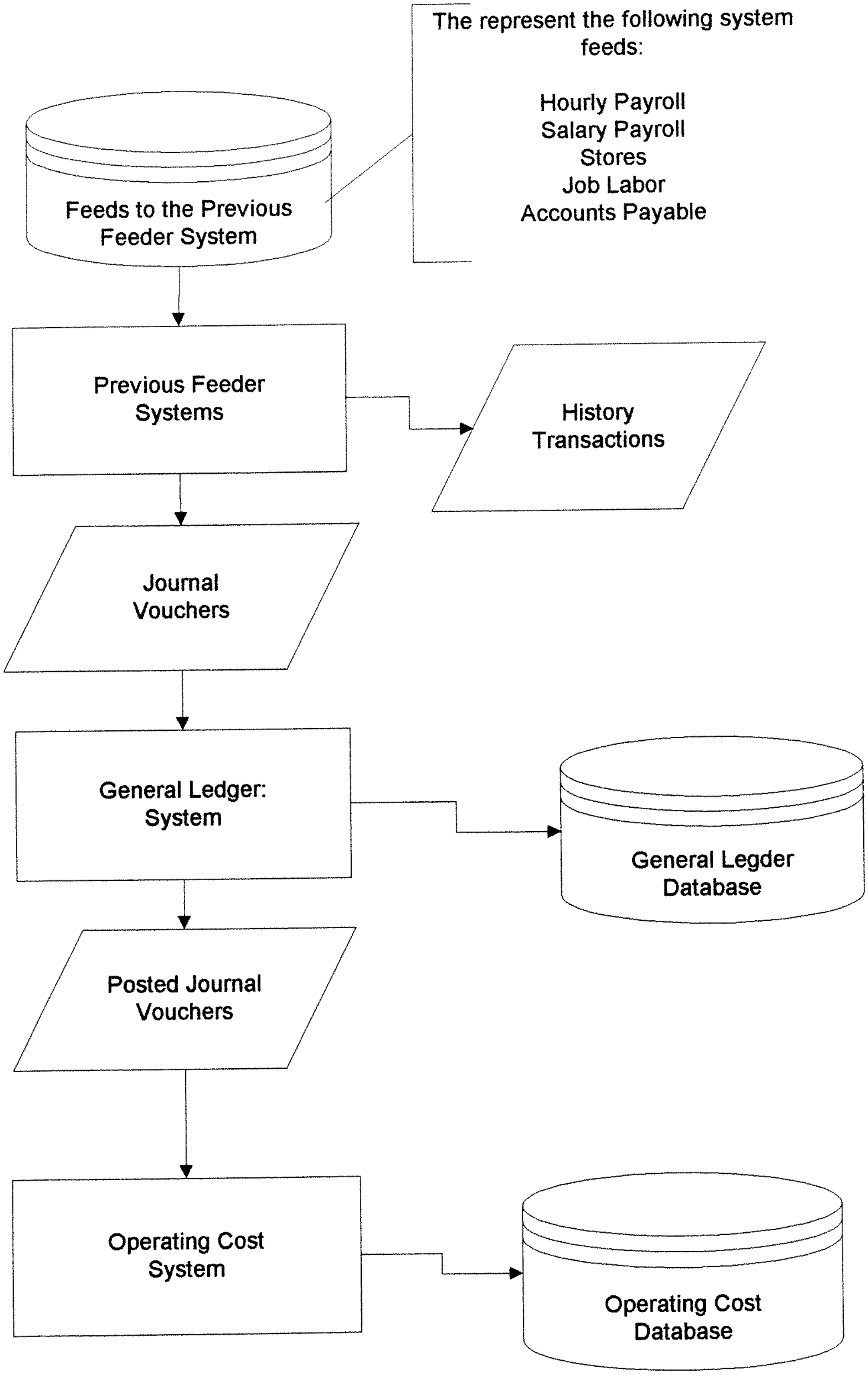


Diagram A

VSAM files. Therefore, because of fiscal constraints, critical financial data and the aging of the financial systems, several sub-systems and batch programs were developed and plan to be developed using conventional flat files to improve the data integrity of the financial systems. However, the future direction of AK Steel Corporation is to develop and/or purchase financial systems that will more effectively and efficiently share data across suprasystem boundaries.

Financial Systems' Problems

Nearly all systems written at AK Steel Corporation were written for a single IBM S390 mainframe machine. As a result, the newer systems, claims, purchasing, and the general ledger systems to name a few, which were added in the last three years to the mainframe machine, had greatly slowed down the performance of all AK Steel Corporation's systems. This was especially true concerning the financial systems' month-end processing. With increased demands being placed on the financial data at month-end, a financial data bottleneck formed because of the additional demands that were placed on the CPU to process the volume of data.

Consequently, the demands of processing this tremendous volume of data through the financial systems had caused two major problems. One of these problems was the increased need to dedicate more CPU time to financial systems to produce financial information within a certain time frame. Another problem involved ongoing data integrity problems, which resulted in time consuming data recovery procedures and the re-running of financial jobs that placed extra demands on the CPU.

Previous System Overview

The earlier constructed Feeder system was considered a sub-system of the General Ledger suprasystem. Its primary uses were to convert transaction journals from several financial systems to the Feeder System and to construct journal vouchers that are submitted and posted to the General Ledger system.

The previous Feeder system consisted of five inputs from financial systems. The five data bridges were the Hourly Payroll, Salary Payroll, Stores, Job Labor and Accounts Payable transactions. These transactions were passed to the former feeder system on a daily or monthly basis. The Job Labor and Accounts Payable transactions were submitted on a daily basis to the previous Feeder system. The transactions that were submitted to the feeder systems on a monthly basis were Hourly Payroll, Salary Payroll and Stores. The previous Feeder system had two primary forms of outputs, which were a journal voucher listing report and the journal voucher records. (See Diagram B)

In addition to the primary processes mentioned, there were several additional processes that were developed to enhance the Job Labor and Accounts Payable Feeders. The additional processes built around the Job Labor feeder were the batch date and the S-order edit processes. The batch date edit process was developed to insure that the correct month was inserted into the Job Labor record. The S-order edit process was developed to properly format the S-order records. Also, the Job Labor feeder contained processes that allowed for manual input of Job Labor rates and special transactions. These processes generated an additional output, which listed the cost of Job

Previous Feeder System

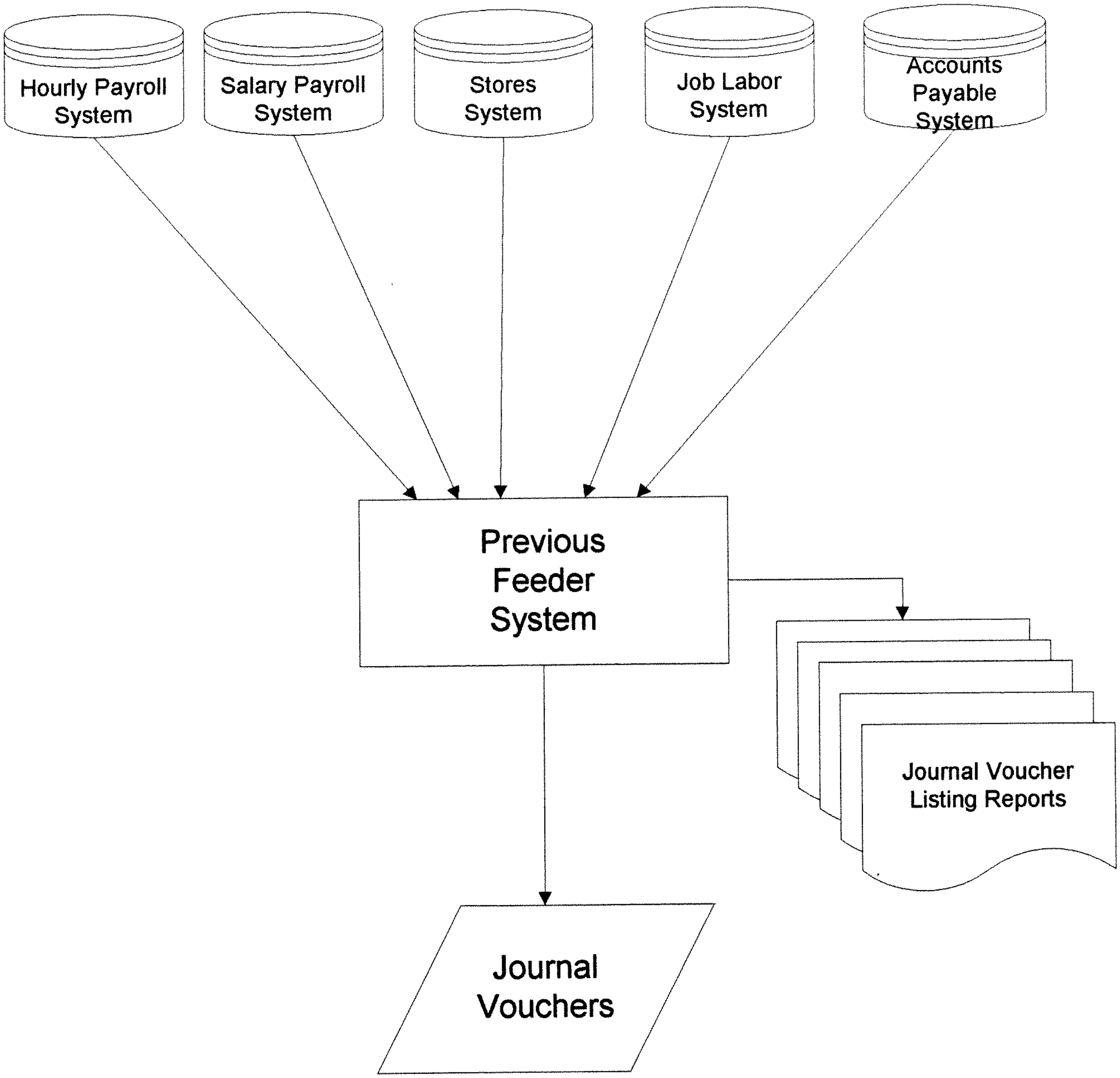


Diagram B

Previous System Overview

Labor associated to a cost center. (A cost center is a predetermined cost facility. e.g. shipping.)

The additional processes built around the Accounts Payable transactions were the validation processes of the daily transaction journals. These processes added an additional level of processing not found in the other Feeders, which attempted to create and validate accounts before generating journal vouchers. The additional outputs created via these processes were the invalid account reports and the stored data of daily processed transactions.

Prior to rewriting the previous Feeder system, the major enhancements to the system were the additional processes added to the Accounts Payable feeder. These enhancements were done two years prior to the rewrite of the new Feeder system. The sponsor (high level end-user) of the new system wanted the entire feeder system to incorporate the same functionality found in the Accounts Payable feeder processes.

Problems of the Previous Feeder System

During month-end processing, the previous feeder system's problems contributed to the bottlenecks found in the increased use of the CPU by the financial applications. The high level end-users of the Operating Cost and General Ledger suprasystems believed that enhancements to the feeder system would eliminate some of the demands on the CPU. The following is a list of problems that led to the rewrite of the feeder system:

- There were no validation processes performed on the Hourly Payroll, Salary Payroll, Stores, and Job Labor journal vouchers. This required the General Ledger system to be used as the primary validation process on journal vouchers. This method of validation proved to be a very costly use of the CPU and an inappropriate use of the General Ledger system.
- Multiple posting runs of the General Ledger resulted because of unedited and/or invalidated journal vouchers. Each posting run took in the range of 30 to 90 clock minutes to complete. It was rather standard procedure to attempt five to seven posting runs before successfully posting all journal vouchers.
- The processing of the Accounts Payable transactions ranged from 30 to 60 clock minutes using the preferred method of validation. However, doing month-end processing, this process significantly delayed the final posting of journal vouchers to the General Ledger.

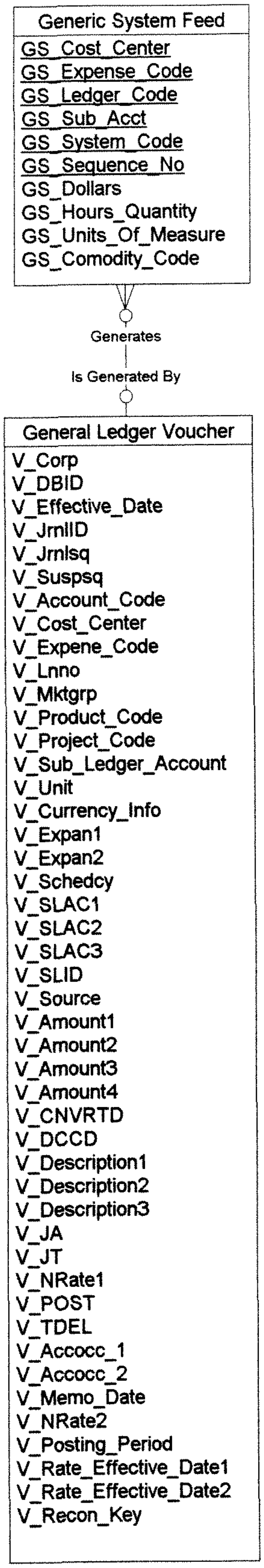
In addition to the problem cause by the increased use of the CPU, other problems of the previous Feeder systems were the following:

- Except for the Accounts Payable transactions, there were no audit trails or history files to reproduce the information submitted to the General Ledger.

Problems of the Previous Feeder System

- The Feeder system's programs were very heavily patched with "hard code" edits. This made enhancements to the programs extremely difficult because it often led to unpredictable outputs.
- End-users were forced to use additional userids and passwords to use the Feeder system. This often caused unnecessary stress to remember a userid and password that were used once a month.

Conceptual Data Model		
Project: Previous Feeder System		
Model : Previous Feeder System		
Author : Michael Menefield	Version: 1.0	7/30/97



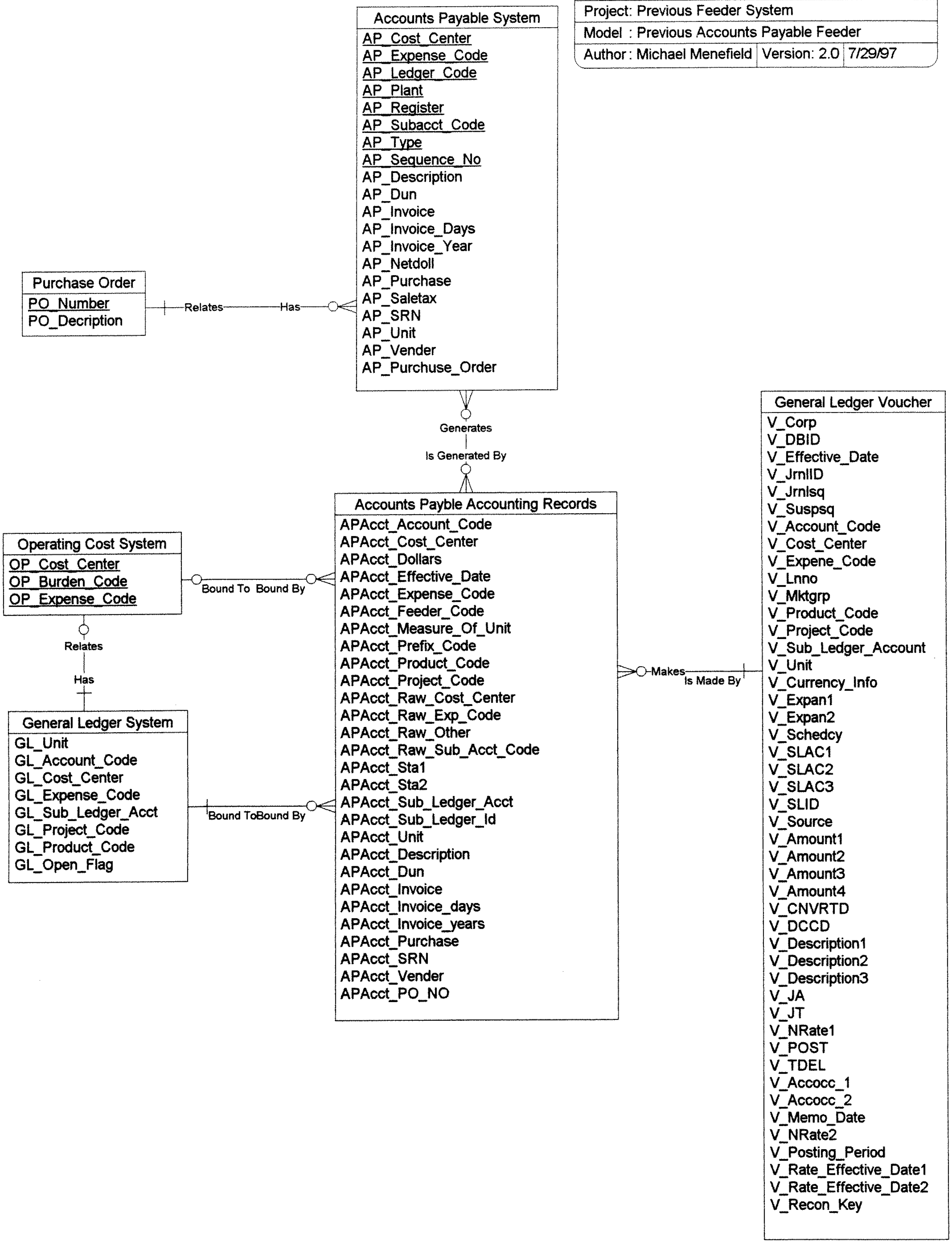
Generic System Feed

GS_Cost_Center
GS_Expense_Code
GS_Ledger_Code
GS_Sub_Acct
GS_System_Code
GS_Sequence_No
GS_Dollars
GS_Hours_Quantity
GS_Units_Of_Measure
GS_Comodity_Code

General Ledger Voucher

V_Corp
V_DBID
V_Effective_Date
V_JrnlID
V_Jrnlsq
V_Suspsq
V_Account_Code
V_Cost_Center
V_Expene_Code
V_Lnno
V_Mktgrp
V_Product_Code
V_Project_Code
V_Sub_Ledger_Account
V_Unit
V_Currency_Info
V_Expan1
V_Expan2
V_Schedcy
V_SLAC1
V_SLAC2
V_SLAC3
V_SLID
V_Source
V_Amount1
V_Amount2
V_Amount3
V_Amount4
V_CNVRTD
V_DCCD
V_Description1
V_Description2
V_Description3
V_JA
V_JT
V_NRate1
V_POST
V_TDEL
V_Accocc_1
V_Accocc_2
V_Memo_Date
V_NRate2
V_Posting_Period
V_Rate_Effective_Date1
V_Rate_Effective_Date2
V_Recon_Key

Conceptual Data Model		
Project: Previous Feeder System		
Model : Previous Accounts Payable Feeder		
Author : Michael Menefield	Version: 2.0	7/29/97



Accounts Payable System

AP_Cost_Center
AP_Expense_Code
AP_Ledger_Code
AP_Plant
AP_Register
AP_Subacct_Code
AP_Type
AP_Sequence_No
AP_Description
AP_Dun
AP_Invoice
AP_Invoice_Days
AP_Invoice_Year
AP_Netdoll
AP_Purchase
AP_Saletax
AP_SRN
AP_Unit
AP_Vender
AP_Purchuse_Order

Purchase Order

PO_Number
PO_Decription

Accounts Payble Accounting Records

APAcct_Account_Code
APAcct_Cost_Center
APAcct_Dollars
APAcct_Effective_Date
APAcct_Expense_Code
APAcct_Feeder_Code
APAcct_Measure_Of_Unit
APAcct_Prefix_Code
APAcct_Product_Code
APAcct_Project_Code
APAcct_Raw_Cost_Center
APAcct_Raw_Exp_Code
APAcct_Raw_Other
APAcct_Raw_Sub_Acct_Code
APAcct_Sta1
APAcct_Sta2
APAcct_Sub_Ledger_Acct
APAcct_Sub_Ledger_Id
APAcct_Unit
APAcct_Description
APAcct_Dun
APAcct_Invoice
APAcct_Invoice_days
APAcct_Invoice_years
APAcct_Purchase
APAcct_SRN
APAcct_Vender
APAcct_PO_NO

Operating Cost System

OP_Cost_Center
OP_Burden_Code
OP_Expense_Code

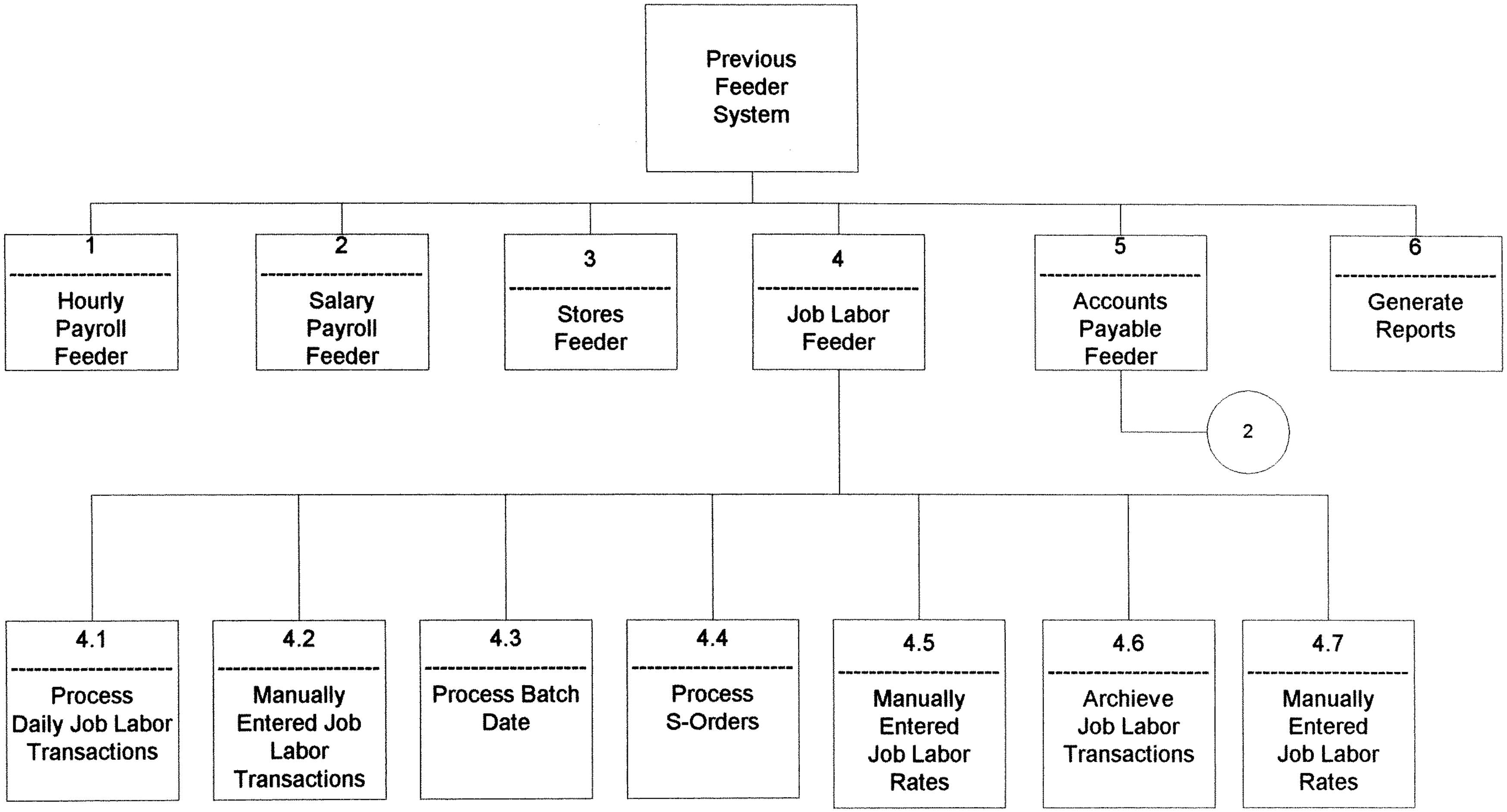
General Ledger System

GL_Unit
GL_Account_Code
GL_Cost_Center
GL_Expense_Code
GL_Sub_Ledger_Acct
GL_Project_Code
GL_Product_Code
GL_Open_Flag

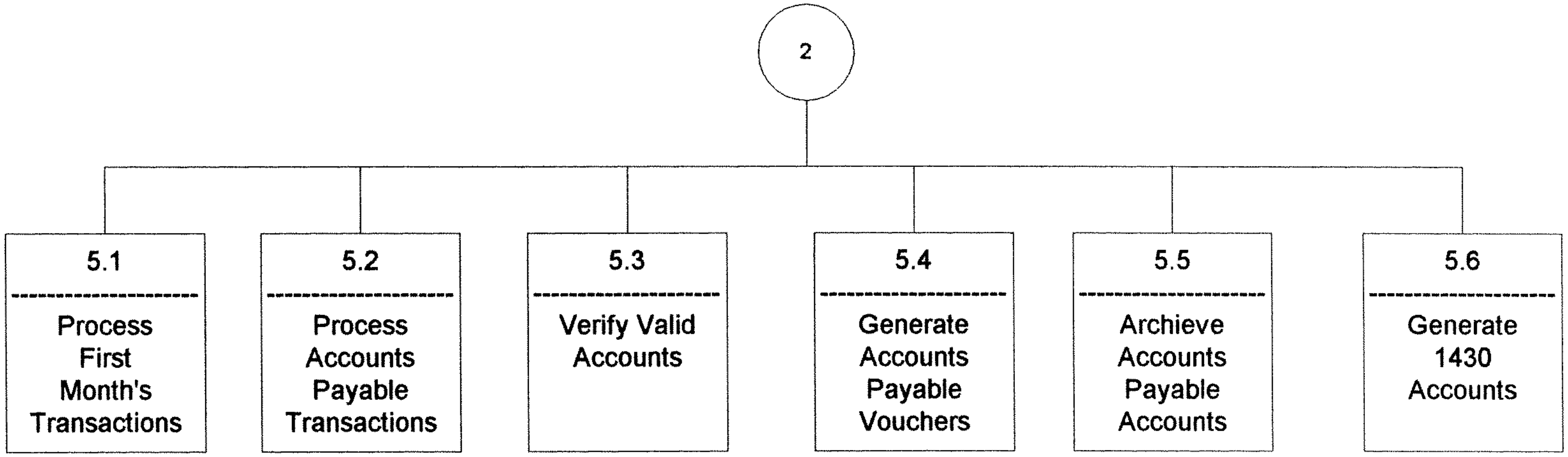
General Ledger Voucher

V_Corp
V_DBID
V_Effective_Date
V_JrnlID
V_Jrnlsq
V_Suspsq
V_Account_Code
V_Cost_Center
V_Expene_Code
V_Lnno
V_Mktgrp
V_Product_Code
V_Project_Code
V_Sub_Ledger_Account
V_Unit
V_Currency_Info
V_Expan1
V_Expan2
V_Schedcy
V_SLAC1
V_SLAC2
V_SLAC3
V_SLID
V_Source
V_Amount1
V_Amount2
V_Amount3
V_Amount4
V_CNVRTD
V_DCCD
V_Description1
V_Description2
V_Description3
V_JA
V_JT
V_NRate1
V_POST
V_TDEL
V_Accocc_1
V_Accocc_2
V_Memo_Date
V_NRate2
V_Posting_Period
V_Rate_Effective_Date1
V_Rate_Effective_Date2
V_Recon_Key

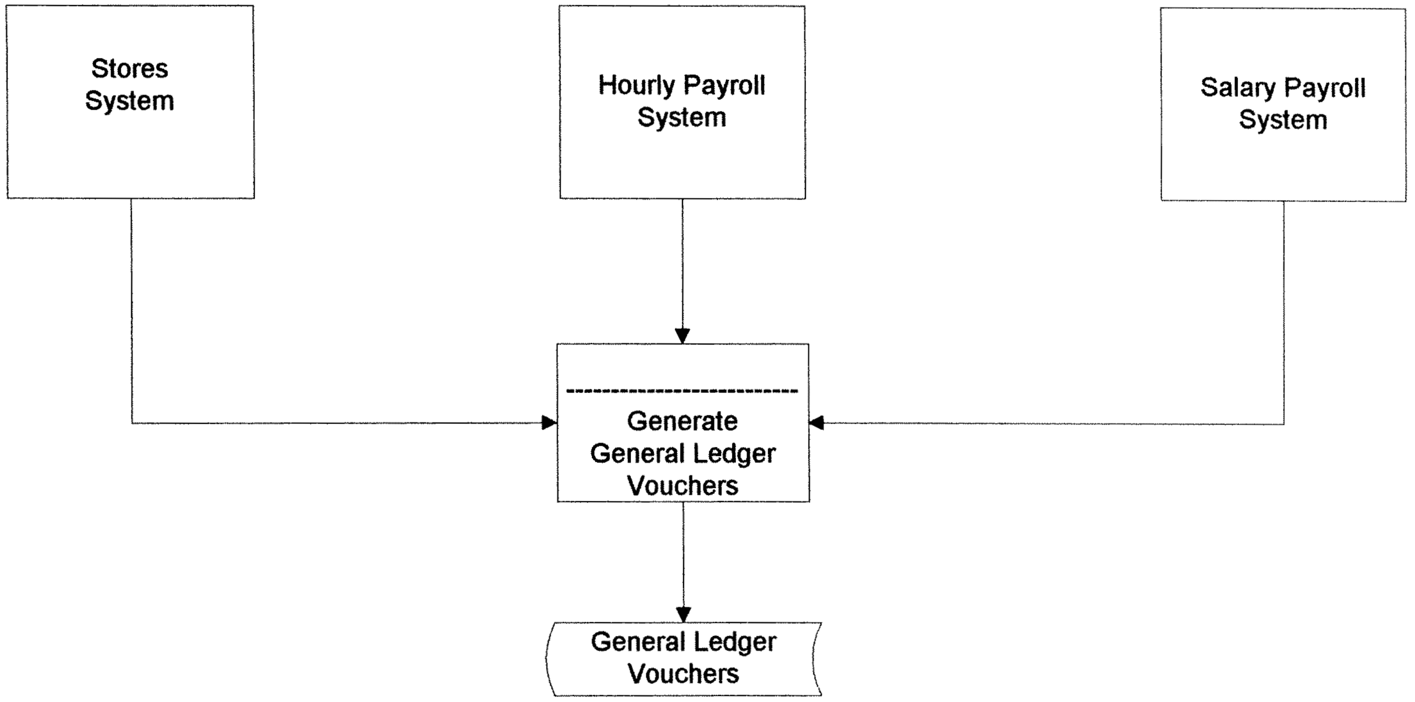
Previous Feeder System
Decomposition Diagram



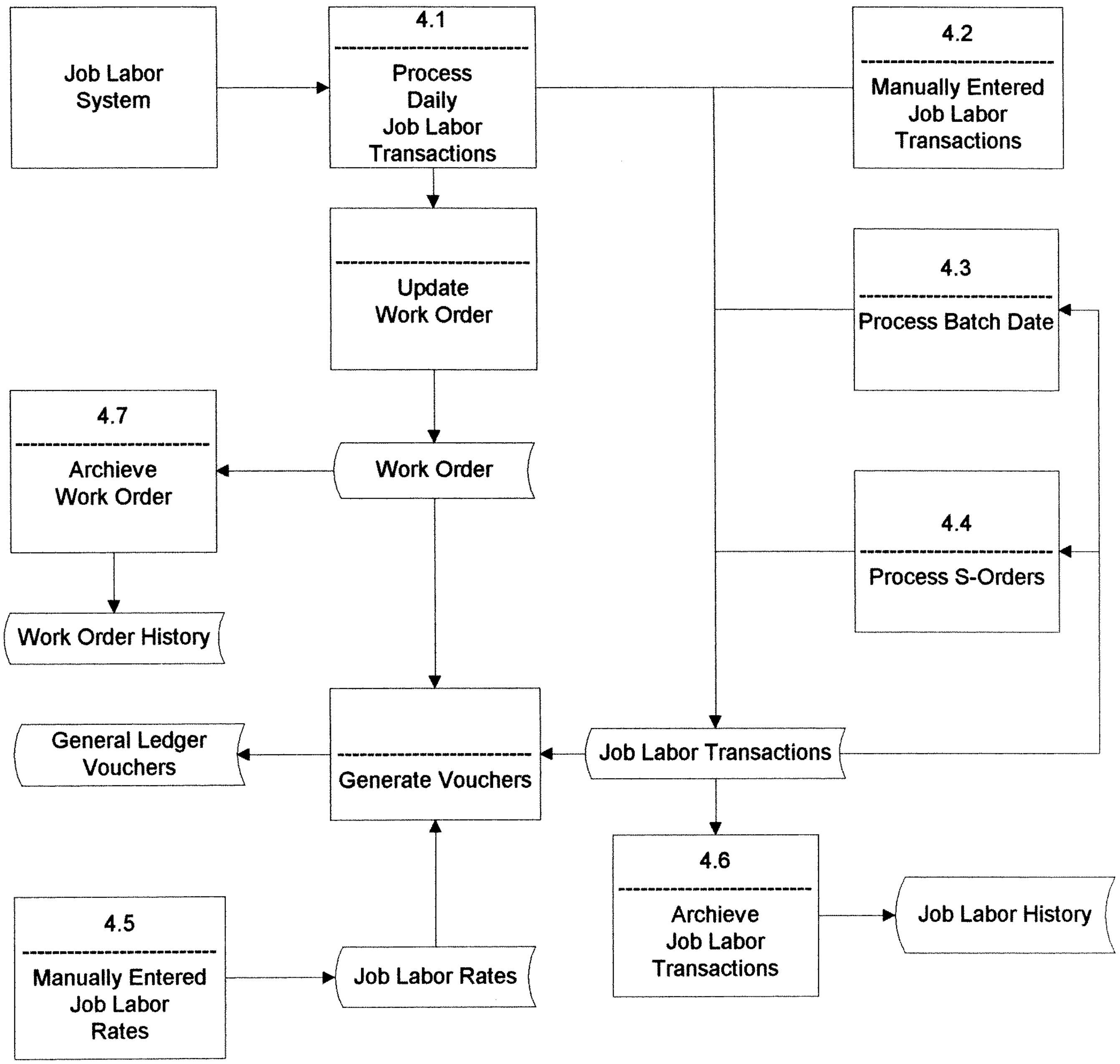
Previous Feeder System
Decomposition Diagram


2
5.1
Process First Month's Transactions
5.2
Process Accounts Payable Transactions
5.3
Verify Valid Accounts
5.4
Generate Accounts Payable Vouchers
5.5
Archieve Accounts Payable Accounts
5.6
Generate 1430 Accounts

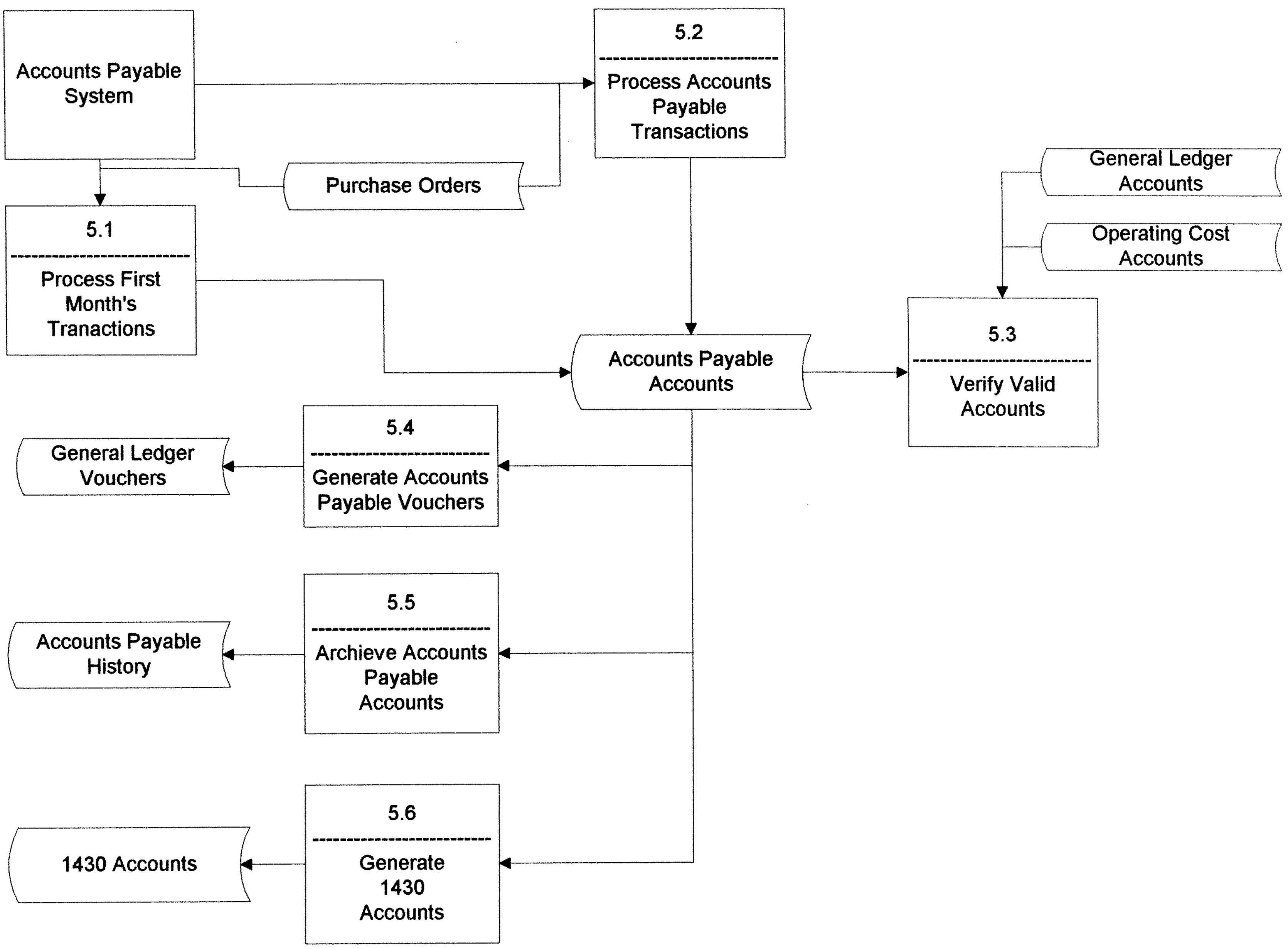
Previous Hourly Payroll, Salary Payroll and Stores Processes



Previous Systems
Job Labor Process Model



Previous Accounts Payable Process Model



Chapter 2

This chapter focuses on the methodology used to develop the new Feeder system. It contains the following sections: project team of the new Feeder system, high-level business requirements, high-level system requirements, test procedures of the new Feeder system, prototype methodologies of the new system and the action plan to synchronize the chart of accounts in the General Ledger and Operating Cost systems. In addition, this chapter contains the generic feeder entity-relationship diagram and generic feeder process model diagram.

Project Team of the New Feeder System

The project team of the new Feeder system consisted of the cost accountant and the system integrator. The cost accountant, a senior accountant in the cost accounting department, was the original developer of the previous Accounts Payable feeder. I was the system integrator named to this project. My previous job responsibility consisted of maintaining the Operating Cost system.

The primary task of the project team was to develop the high-level system requirements and develop the high-level system specifications of the new Feeder system. Also, the cost accountant was charged with addressing any question that I, the system integrator, had concerning the requirements and functionality of the new Feeder system.

I had the responsibility of developing and refining the system requirements and specifications as well as developing the remaining components of the new feeder system. All of the components used to develop the new Feeder system will be discussed throughout the remainder of this paper.

High Level Business Requirements

After consulting with the high level end-users and the project team, the following list of business requirements were identified:

- Eliminate the need to re-run the General Ledger System's posting process because of invalid accounts being loaded into the journal vouchers from the Feeder system.
- Integrate the new Feeder system's menu navigational system with each of the individual feeder menus.
- Develop the new Feeder system with a flexible account conversion process, which does not require recompilation of the program to add new account conversion logic.
- Reduce the real clock time of the feeder processes to run 30 minutes or less under normal conditions.
- Eliminate the additional userid and password that the end-user needs to access the feeder system.

High Level System Requirements

After consulting with the high level end-users and the project team, the following list of system requirements were developed. These system requirements were to be addressed in the design, development and implementation of the new Feeder system:

- Develop a yearly history file for each of the five feeders: Hourly Payroll, Salary Payroll, Job Labor, Stores and Accounts Payable Feeders
- Validate every journal transaction against the General Ledger and Operating Cost Systems' Chart of Accounts
- Develop a standardized cost accountancy file format to be utilized by the each of the five feeders
- Develop a standardized report listing and error report for the each of the five feeders
- Develop a standardized suspense file to capture journal transactions that do not pass the account validation process for each of the five feeders
- Develop a menu system that will assist the end-users in processing transaction journals
- Develop each feeder to consist of four functions: retrieve, edit, load and archive functions
- Develop a process to back-up all daily batch submitted transaction journals

Test Procedures of the New Feeder System

The test methodology of the new feeder system covered the testing of individual programs to the entire Feeder system. Below is a list of procedures to follow prior to implementing the new program:

1. The project team will perform the Alpha test, using test data, on the new program to verify it meets prescribed program specifications. If the program fails the Alpha test, make appropriate modifications to the program. Continue to perform this test on the new program until it successfully meets specification requirements.
2. The project team will perform the Beta test, using prior month's production data, on the new program to verify it meets the prescribed specifications. If the program fails the Beta test, make appropriate modifications to the program and perform the Alpha test. Make modifications to the Alpha test data and the program specification requirements.
3. The project team will perform the Beta test on the process or processes containing the new program. Verify the test output for accuracy and correctness. If the test fails then determine the inappropriate output, make modifications to the new program and perform the Alpha test on the modified program.
4. The project team will perform the Beta test on all of the processes of the system that will be affected by the new program. Verify the accuracy and correctness of the output. Examine the Feeder system's response time to determine appropriate performance of the system. If the Beta test fails then make appropriate modifications to the program and program specifications. If the beta test returns inappropriate response time, re-evaluate the program specifications. If necessary, re-write the feeder process.

Test Procedures of the New Feeder System

5. The project team will allow the end-users to evaluate the modified process. If the end-users approve the results of the process, place the new program into the production job stream. This will allow the new program to be used by the feeder system. If the end-user does not approve of the results, make appropriate changes to the program specifications. If inappropriate, re-write the feeder process.

Prototype Methodologies of the New Feeder System

Several methods of prototyping were used in developing specifications and designing the new Feeder system. The first method of prototyping was to develop screens using the traditional screen sketch drawings. This enabled the user to physically see the concepts discussed in developing the Feeder system. There were two purposes in using the aforementioned method. The first purpose was to encourage the end-users' to take ownership of the new Feeder system. The cost accountancy end-users were aware that there were plans being developed to create a new Feeder system; however, past plans never materialized into a new Feeder system. The second purpose was to gather more system specifications. Most of the end-users were reluctant to have any involvement in a system that might not benefit them. It was hoped that the screen drawings might provide some type of assurance that a system was to be developed to assist them on their jobs. Consequently, the end-users were more apt to participate in system specification exercises.

The second method of prototyping was to develop the new Feeder system's navigational menus. The menu screens were developed using the screen drawings and other specifications gained from the first prototype method. This prototype was to allow the end-user to easily maneuver through the menu screens to select various feeder options. The navigational menus were created in the interpretative program language called CLIST. This language provided for the screen painting of menus and other interactions needed in batch programming. Also, CLIST can perform TSO (Terminal Sharing Option) functions as well as call TSO commands. CLIST is the language of choice at the AK Steel site when developing batch interactive screen programs.

Prototype Methodologies of the New Feeder System

The second prototype of the new Feeder system contained the four prescribed functions of the Hourly Payroll Feeder. The four prescribed functions were:

- Retrieve the transaction journals to the Feeder system
- Edit the transaction journal vouchers
- Load the processed journal vouchers to the General Ledger system
- Archive the monthly Hourly Payroll transaction journals.

The second prototype gave the user the ability to maneuver through the main Feeder system's screen and the secondary menu screens.

Several prototype modifications were made to the Hourly Payroll Feeder. These modifications were made until a fully tested Hourly Payroll feeder was developed. As a result, the Hourly Payroll feeder sparked enthusiasm for the new Feeder system. This rapid prototype approach was used to produce the other four feeders, Salary, Stores, Job Labor and Accounts Payable Feeders.

Action Plan to Synchronize the Chart of Accounts

The chart of accounts of the General Ledger and the Operating Cost systems are extremely important components of the new Feeder systems' validation process. The validation process compares the feeders' journal vouchers with the valid account codes found in the chart of accounts of the previously mentioned systems. Because of the importance of the validation process, discovered in the development of the Hourly Payroll prototype, I realized that I needed to develop an action plan to synchronize the aforementioned systems' chart of accounts. As a result, the performance of the validation process was enhanced when the General Ledger System's chart of accounts was synchronized with the Operating Cost System's chart of accounts. After questioning the end-users, I found out that the General Ledger and Operating Cost system's chart of accounts were out of synchronization for a number of years. The following list of tasks was the action plan to synchronize the chart of accounts:

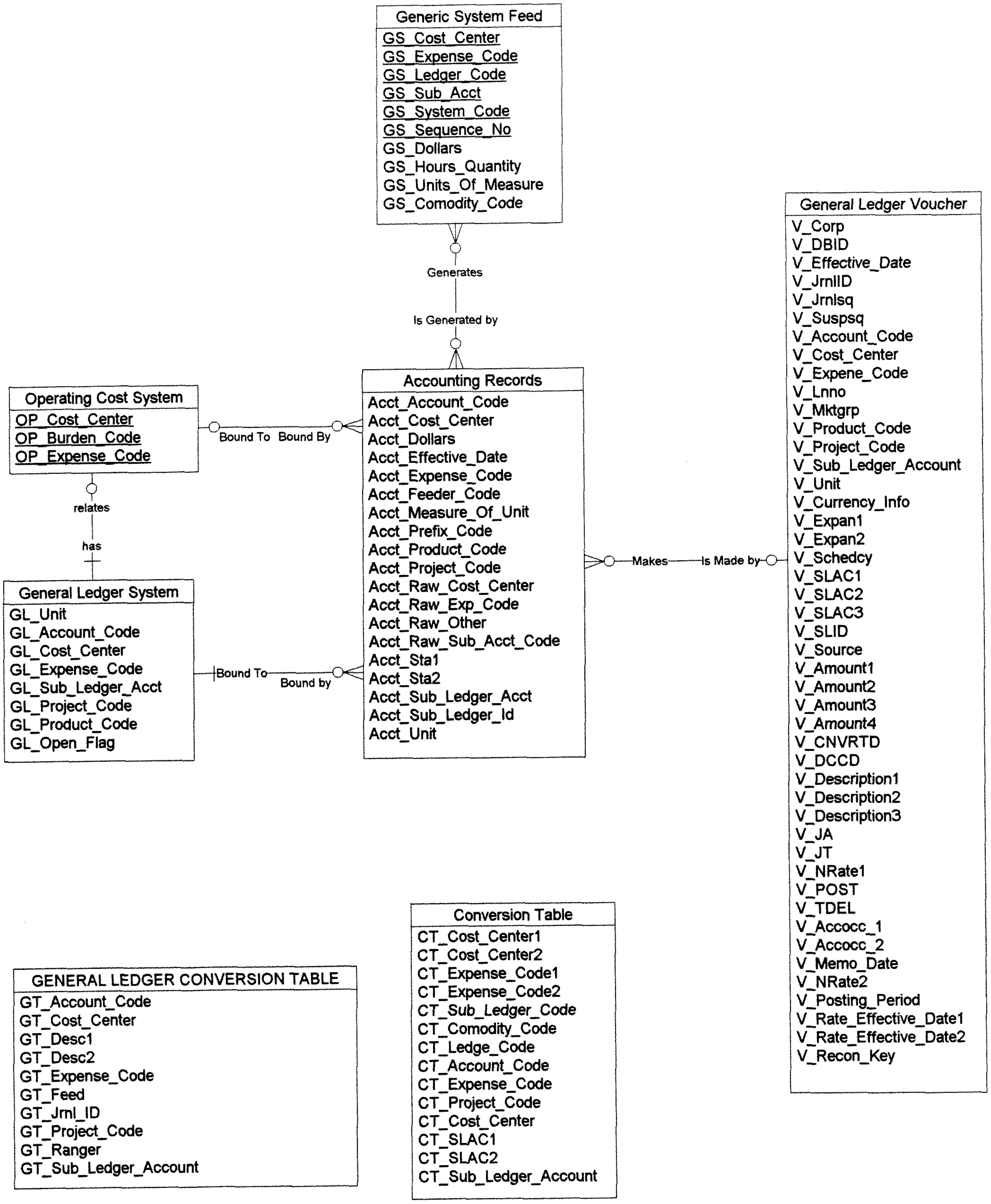
1. Compare the General Ledger and Operating Cost Systems' chart of accounts; create a report to show the accounts designated with a unit code of 28 (unit code 28 represents Middletown Work accounts), in the General Ledger System which are not found in the Operating Cost system
2. Create a file containing the accounts which should be closed or deleted out of the General Ledger System's chart of accounts; use this file to develop a batch procedure to merge the accounts containing a non-zero balance, which have been picked to be closed or deleted, with a valid General Ledger account; use the file containing accounts which should be closed and deleted out of the General Ledger to develop a batch procedure to close these account in the General Ledger System

Action Plan to Synchronize the Chart of Accounts

3. Compare the General Ledger and Operating Cost Systems' chart of accounts to verify that the two system's chart of accounts are synchronized. If the charts of accounts of the two systems are not synchronized, perform all the tasks again.

After the chart of accounts of the General Ledger and the Operating Cost systems matched in regard to the Middletown Works accounts, the performance and validation accuracy of the new Feeder system will depend on the end-users' ability to maintain the chart of accounts. The end-users are required to add, delete and close appropriate Middletown Works accounts in both systems to maintain the synchronization of the aforementioned systems' chart of accounts. Because the General Ledger and the Operating Cost systems are interdependent on each other, only maintaining one system's chart of accounts will degrade the performance of the new Feeder system.

Conceptual Data Model		
Project: New Feeder System		
Model : Generic System Feed		
Author: Michael Menefield	Version: 1.0	6/27/97



Generic System Feed
GS_Cost_Center
GS_Expense_Code
GS_Ledger_Code
GS_Sub_Acct
GS_System_Code
GS_Sequence_No
GS_Dollars
GS_Hours_Quantity
GS_Units_Of_Measure
GS_Comodity_Code

Operating Cost System
OP_Cost_Center
OP_Burden_Code
OP_Expense_Code

Accounting Records
Acct_Account_Code
Acct_Cost_Center
Acct_Dollars
Acct_Effective_Date
Acct_Expense_Code
Acct_Feeder_Code
Acct_Measure_Of_Unit
Acct_Prefix_Code
Acct_Product_Code
Acct_Project_Code
Acct_Raw_Cost_Center
Acct_Raw_Exp_Code
Acct_Raw_Other
Acct_Raw_Sub_Acct_Code
Acct_Sta1
Acct_Sta2
Acct_Sub_Ledger_Acct
Acct_Sub_Ledger_Id
Acct_Unit

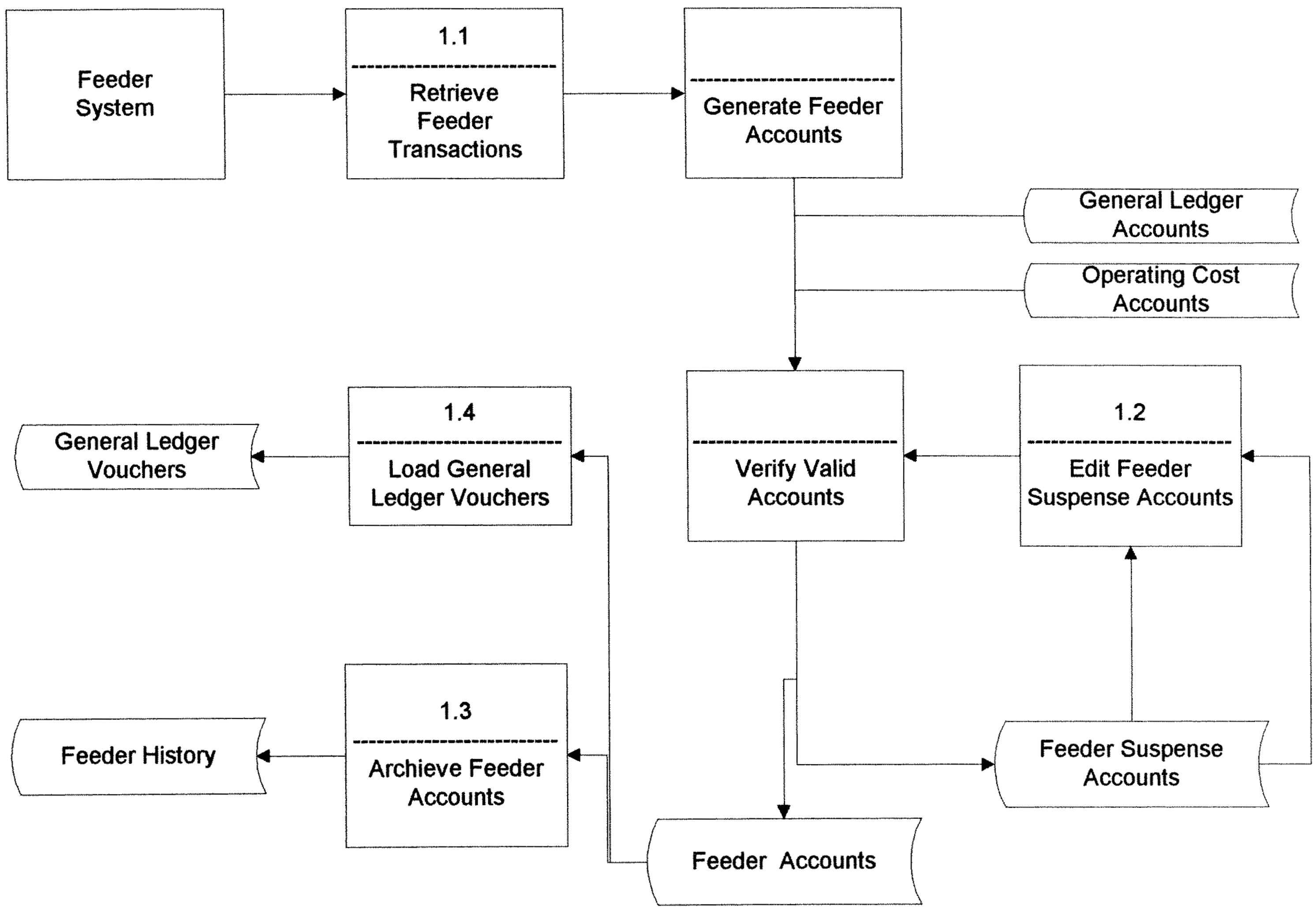
General Ledger Voucher
V_Corp
V_DBID
V_Effective_Date
V_JrnlID
V_Jrnlsq
V_Suspsq
V_Account_Code
V_Cost_Center
V_Expene_Code
V_Lnno
V_Mktgrp
V_Product_Code
V_Project_Code
V_Sub_Ledger_Account
V_Unit
V_Currency_Info
V_Expan1
V_Expan2
V_Schedcy
V_SLAC1
V_SLAC2
V_SLAC3
V_SLID
V_Source
V_Amount1
V_Amount2
V_Amount3
V_Amount4
V_CNVRTD
V_DCCD
V_Description1
V_Description2
V_Description3
V_JA
V_JT
V_NRate1
V_POST
V_TDEL
V_Accocc_1
V_Accocc_2
V_Memo_Date
V_NRate2
V_Posting_Period
V_Rate_Effective_Date1
V_Rate_Effective_Date2
V_Recon_Key

General Ledger System
GL_Unit
GL_Account_Code
GL_Cost_Center
GL_Expense_Code
GL_Sub_Ledger_Acct
GL_Project_Code
GL_Product_Code
GL_Open_Flag

GENERAL LEDGER CONVERSION TABLE
GT_Account_Code
GT_Cost_Center
GT_Desc1
GT_Desc2
GT_Expense_Code
GT_Feed
GT_Jrnl_ID
GT_Project_Code
GT_Ranger
GT_Sub_Ledger_Account

Conversion Table
CT_Cost_Center1
CT_Cost_Center2
CT_Expense_Code1
CT_Expense_Code2
CT_Sub_Ledger_Code
CT_Comodity_Code
CT_Ledge_Code
CT_Account_Code
CT_Expense_Code
CT_Project_Code
CT_Cost_Center
CT_SLAC1
CT_SLAC2
CT_Sub_Ledger_Account

Generic Feeder Process Model



Chapter 3

This chapter delineates the learning experiences gained in using several phases of the System Development Life Cycle (SDLC) on the new Feeder system. It contains the following sections: the work journal of the project and the problems in using various phases of the System Development Life Cycle.

Work Journal of the New Feeder System

The amount of time I was assigned to the new system varied, throughout its completion. The following list contains the percentage of the time I was to dedicate on the design, the development and the implementation of the new Feeder system.

- In March 1996, 25% of my workweek was dedicated to the project
- In May 1996, 50% of my workweek was dedicated to the project
- In June 1996, 60% of my workweek was dedicated to the project
- In September 1996, 75% of my workweek was dedicated to the project

Until the completion of the project, 75% of my workweek remained dedicated to the project

Work Journal of the New Feeder System

March 1996

- Assigned and met to the project team
- Reviewed the Accounts Payable transaction processes
- Examined the other feeder processes
- Defined the High Level Business Requirements
- Divided the New Feeder system into five feeders, Hourly Payroll, Salary Payroll, Stores, Job Labor and Accounts Payable Feeders, to develop the system in phases
- Developed the new Feeders system's high level specifications
- Developed the new Feeders system's hierarchy chart
- Developed file relationship diagrams
- Developed a simple process model
- Developed a structure flowchart
- Developed a quick prototype to gather other system specifications

April 1996

- Designed and developed the Hourly Payroll Feeder
- Designed and developed the Salary Payroll Feeder
- Designed and developed the Job Labor Feeder
- Examined the General Ledger and Operating Cost systems' chart of accounts; developed an action plan to synchronize the chart of accounts of the General Ledger and the Operating Cost

Work Journal of the New Feeder System

Systems

May 1996

- Tested and implemented the processes of the Hourly and Salary Payroll Feeders
- Began testing the processes of the Job Labor Feeders
- Deleted and closed accounts in the General Ledger and the Operating Cost system to synchronize the databases' chart of accounts

June 1996

- Had problems loading the journal vouchers of the Hourly and Salary Payroll Feeders to the General Ledger, developed a new General Ledger interface testing procedure, reviewed the General Ledger interface layout
- Tested and implemented the modified Hourly Payroll and Salary Feeder layout
- Tested and implemented the processes of the Job Labor Feeder
- Designed and developed the error correction process of the Job Labor Feeder
- Deleted and closed accounts in the General Ledger and the Operating Cost system to synchronize the databases' chart of accounts

July 1996

- Designed, developed, tested and implemented the processes of the Stores Feeder

Work Journal of the New Feeder System

- Changed the Stores and Job Labor Feeders to load detail journal vouchers to the General Ledger interface file
- Separated the Stores Feeder into Stores, Operating Supplies and Spares accounts
- Implemented the modifications of the Stores and Job Labor Feeders

August 1996

- Had problems loading the volume of detail journal vouchers to the General Ledger interface from the Stores and Job Labor Feeders, began a process to summarize the detail journal vouchers for the Hourly Payroll, Stores, Job Labor and Stores Feeders
- Designed, developed, tested and implemented the Job Labor Report, Department Hours Report, S-Order Process and Date Process of the Job Labor Feeder

September 1996

- Designed and developed the processes of Accounts Payable Feeder
- Designed, developed, tested and implemented the Job Labor Correction and Weekly Reports
- Began testing the processes of the Accounts Payable Feeder

October 1996

- Designed, developed, tested the re-edit process of the Stores Feeder
- Implemented the Stores Feeder's re-edit process

Work Journal of the New Feeder System

- Continued testing the processes of the Accounts Payable Feeder

November 1996

- Designed, developed, tested and implemented a Cost Center Summary Report for the Job Labor Feeder
- Continued testing the processes of the Accounts Payable Feeder
- Implemented the processes of the Accounts Payable Feeder
- Completed the implementation of the five feeders: Hourly Payroll, Salary Payroll, Stores, Job Labor and Accounts Payable Feeders; met project deadline

Problem in Specifying Requirements

In specifying requirements, my challenge was to definitively define the functional and non-functional requirements of the new system for the purposes of establishing the scope of the project. These requirements formed the contractual agreement between the project team and the end-user(s). The non-functional and functional requirements were developed from analyzing the current system's functionality, effectiveness, presentation and usability. Also, the requirements were gathered from the sponsor, a high level end-user, and the brainstorming activities of the project team. In the following paragraphs, I will discuss the four problem areas, which I encountered in defining the specification requirements of the new system.

The first problem that I faced was the over-controlling role of the sponsor, the cost accountant manager of the Middletown Works sites. In evaluating the successful projects at AK Steel Corporation, I found that an active company sponsor was the key to its success. The sponsor determined the success of the project with his or her ability to assign appropriate resources to the project. I discovered that a good project often failed or was canceled due to its lack of resources because it lacked a sponsor. I was fortunate to have a number of sponsors on the project that desired to see its completion. However, the amount of sponsor's control over the project drastically affected the deliverables of the system.

In developing the new Feeder system, the sponsor took a very active role in specifying the requirements of the project. At the conception of the project, the sponsor placed limitations on the project team. In defining business and system requirements, the sponsor directed the project team not to involve anyone else in the requirement development except for him. This stringent directive posed the problem of gathering information from all of the appropriate end-users. With this limitation on the resources, the requirement specification was limited to defining only the new Feeders' system common functions. The more definite specifications of the system were to be deferred to the use of the system's prototype.

The second problem I faced was to analyze, design, construct, test and implement the new Feeder system by December 1, 1996. This problem caused me to manage the project to the date, which compelled me to shorten the phases of the System Development Life Cycle. The challenge to the project team throughout the project was to produce results expeditiously, which increased the temptation to stop analyzing the current and new systems and to begin coding. Because of the need to produce results quickly, I chose to develop the new Feeder system using the rapid prototype approach to system development.

The third problem I encountered was the project's team methodology of gathering specification requirements. There was often a struggle to resort to the status quo methodology of developing a system, which was to change the functionality of the current system that did not work. The status quo methodology results in nothing more than a system enhancement. With the questionable status quo methodology waiting to be used by others in this stage of system development, I was even the more challenged to develop plausible solutions for the system and to develop definitive system requirements.

Problem in Specifying Requirements

The fourth problem I encountered was how to limit the scope of the project. With the project team limited to developing generic system specifications, I believed that the most appropriate course of action was to concentrate on definitively defining the specification requirements gathered. Consequently, this approach reduced the amount of rework performed on the system after implementation of the prototype. Due to the generic system requirements, the true scope of the project was not known until after the implementation of the prototypes. While this was the true purpose of the prototype, no adjustments were made to the system deadline of completion, after the true scope of the system was identified.

Problems in Designing Specification of the System

In designing specifications of the new system, I took the functional and non-functional requirements of the system and developed a generic entity-relationship model and a generic process model to help capture the essence of the system. I used the entity-relationship modeling to show and understand the relationship between the new Feeder system's entities. I utilized the process modeling to capture the suggested functionality of the new Feeder system. The problems I had in developing specifications can be categorized into three problem areas. These problem areas were the specified time restraints, the simplistic entity-relationship modeling and the simplistic process modeling. The following paragraphs will discuss each of these problem areas.

The first problem that I noted in this section was the specified time restraints placed on the development and implementation of the new Feeder system. Because of this restraint, I chose to reduce the amount of time allocated to perform analysis on the current system and the new system. In my effort to define and develop system specifications, I took the risk of standardizing the functionality of each of the five feeder processes, Hourly, Salary, Stores, Job Labor and Accounts Payable, of the new Feeder system. Consequently, all of the brainstorming exercises focused on finding ways to standardize the specifications, instead of producing plausible solutions or strategies. My direction of developing specification concentrated on acquiring generic aspects of the system to develop a successful system skeleton. If specialized processes were required for each of the aforementioned feeder, it would be addressed during the development and implementation of that feeder's prototype. As a result of the time restraint, only a few solutions were even explored.

Problems in Designing Specification of the System

The second problem was in the development of the simplistic process models. The process models treated all five feeders in the same way. Each process model only included the four basic processes. These processes were the retrieval, edit, load and archive processes. These processes were used to develop a vanilla modeling of the system. The specific or special processes of an individual feeder system were deferred to the implementation of the prototypes.

The third problem was in the development of the simplistic data models. Once again, each of the feeder's entity-relationship diagrams was developed with the objective to standardize the models. Hence, all entities were standardized. Consequently, each of the feeders data model was developed using the following six entities: transaction journals, feeder accounts, General Ledger chart of accounts, Operating Cost's chart of accounts, feeder history accounts and General Ledger accounts. The focus of the data model was to form common relationships between each of the feeders. Once again, the unique entities of the individual feeder system were to be analyzed, discovered and designed during the implementation of the prototype. Normally, this would not be a problem, however, given the time restraints, I was focused to recognize and to develop each of the five feeders unique feature in a small and unknown amount of of time.

As mentioned in the previous sections, the purpose for developing the specifications was to develop a vanilla system, which consisted of the essential functions of the system. The other aspects of the system were left to be the discovered using the rapid prototype approach to system development.

Problems in Designing the System

After the capturing of the system requirements and developing of the system specifications, my approach deviated from my class training to use a series of rapid prototypes to develop the component parts of the system. This change was due to the time restraints being place on the project. Afterwards, using the rapid prototype method, I choose to systematically use the component parts, which were developed to design and implement several different levels of a system prototype. The developments of the prototypes were to aid the end-user(s) in specifying definitive business and system requirements. Also, the prototypes were used to explore system outputs such as reports, listings and files. In addition, the prototypes would examine the use of processes and the usability and effectiveness of the navigational menu.

The problems I had in designing the system can be grouped into two problem areas. These problem areas were the pitfalls of prototyping and the designing of generic file to be used by all of the five feeders. The following paragraphs will address each of these problem areas.

The first problem was the pitfalls of prototyping that resulted from a generic approach in gathering requirements and developing system specifications. As a result, the prototype was used to discover the unique attributes of the five feeders system: Hourly Payroll, Salary Payroll, Job Labor, Stores and Accounts Payable Feeders. At this stage in the system development life cycle, the five feeders were handled individually to determine the unique aspects of their processing needs. This approach to system analysis and design added more complexity to the system's design. However, as a result of prototyping the systems, the end-users were able to further explore and envisioned other possible uses of the new systems, which increased the original scope of the project. While the rapid and system prototypes were valuable tools, I found that the end-users would often accept the first feasible solution to solve the problem while foregoing the search for alternative solutions. Throughout the project, I found that the most expeditious solutions often were written because of missing or changing user requirements. Therefore, I believed that if more attention was given to finding good solution that the time use to rewrite the solution could have been better spent. For the most part, the problems that were found in using the prototypes were anticipated.

The next problem that I faced related to designing a generic file format to be used by each of the five feeders: Hourly, Salary, Stores, Job Labor and Accounts Payable. This caused a problem in file design because all of the five feeders varied in the data being received from outside systems. Consequently, I was required to redesign the generic file format every time an additional feeder was added to the system because of the changes in the new Feeder system's scope. Eventually, a new file format was agreed upon which defined 15 standardized fields in every file and made 250 characters the standard length of the files. In addition to the new file format, more processes were developed to capture the supplementary data that needed to be place in the newly, formatted standardized files.

Problems in Constructing the System

After prototyping the system, the task was to construct the system. The tools and techniques I employed in constructing the system were a top-down system development approach. This included the use of hierarchy charts and structured flowcharts. This was a requirement of the sponsor because he desired to better understand the data flow of the feeders. In addition, I used the structured approach to system development coding that required a module to contain only one function/task when possible. I used the structure flowcharts to identify the function/task needed to be code in the module.

In constructing the system, I was limited to using the programming languages of CLIST, COBOL, PL1 and SAS. Consequently, I coded the navigational menu screens in the interpretative language called CLIST and used PL1 and SAS programming languages to code the remainder of the batch system.

The problems I had in constructing the system can be grouped into two problem areas. These problem areas were the development of "hard-code less" programs and the use of "ego" focused programming. The following paragraphs will discuss each of these problem areas.

The first problem that I faced was the task of developing programs in which the use of hard-code edits was to be avoided. Hard-code edits were edits written into a program, which required a program change to alter its affect in a program. For example, it is not uncommon to find a hard-code edit that changes the value of a cost center field from the value of '2000' to '0100'. With this use of hard-code edits, often-reusable programs were turned in very cryptic coded programs. I found that hard-code edits were often used to bandage a solution to a programming problem. This bandage approach to programming was frequently used in the maintenance cycle of a system; however, extended use of this approach has caused maintenance nightmares. It was my intent to limit the number of programs that required the use of hard-code edits. Therefore, I developed the parsing programs' logic external to the program in a rule-based table. This allowed for the ease to change of rule of the parse logic without having to recompile the program. In my effort to write programs that do not use hard-code edits, several rewrites to programs were required to limit or eliminate the use of hard-code edits.

The next problem that I faced resulted from my initial decision of not utilizing "egoless" programming approach in constructing the new Feeder system. Because I was solely responsible for the system's development, I took extra precautions in designing the layout of each program. It became no longer a new financial system but my personal financial system in which I was given the sole responsibility of programming.

I made it a point to carefully develop and adhere to a particular programming style and solution approach. This programming style ensured that all programs had standard features. Consequently, the layout of all the programs had a similar modular component. For example, all programs share in the use of similar modular formats such as all navigational programs defined the function keys in the last modular. This format style provided for ease in maintaining and debugging the programs. Consequently, even workable solutions of the previous system were rewritten to conform to my programming style. This stylistic demand lengthened the implementation of the completed system; however, it standardized the programs used in the new Feeder system.

Problems in Testing the System

After the construction of the system was complete, I began testing the individual components of the feeders. These tests included evaluating individual programs, the process containing the programs, the feeder associated with the process and the entire Feeder system. The test procedure was to insure that the logic of the function(s) worked as specified. In addition, the end-users had an opportunity to test the system to insure its correctness. Consequently, the questions to address were "Does the system work and does the system work right?"

In developing the system test, I was faced with two major problems. These problems were the development of a test procedure for the integrated systems and a volume test procedure for the integrated systems. The following paragraphs will address these two problems.

The first problem resulted from not developing or planning the appropriate test for the integrated systems. The output files of the Feeder system comprised the General Ledger interface file, the primary input of the General Ledger posting process. While there were many tests performed on the feeder system and the system's outputs, I did not develop an adequate procedure plan to test the interface file of the General Ledger System. Consequently, new procedures were added to the test plan for purposes of testing what has been overlooked..

The second problem resulted from not developing a volume test of the integrated systems. This problem materialized when there was over a 1000% change in the volume of Job Labor and Stores transaction voucher records that were passed to the interface file. This change resulted in crashing the General Ledger system during the critical production cycle of the financial systems processing. Consequently, a volume test was added to the systems' test procedures. This test would ensure that major changes in the volume of data would be acceptable changes to the current operation of both the General Ledger and Operating Cost systems.

Problems in Implementing the System

After the successful testing of a program, process and feeder, the task of implementation was to place the function into a production set of libraries. The method of implementation was to place programs into production by gradually phasing-in the tested code. The sponsor had agreed that I would develop, design, construct, test and implement the Feeder system in five predetermined component feeders: Hourly Payroll, Salary Payroll, Job Labor, Stores and Accounts Payable. In addition, the sponsor agreed to phase-in each of the five feeders separately. Therefore, the entire Feeder system was planned to take five major phase-in processes.

Often, the problems in implementing the system were caused because the modified and changed programs were not appropriately placed in production. This resulted from testing and getting end-users approval to place the program(s) into production; however, I did not always perform this task in a timely matter. Consequently, some system changes failed because the modified program was not found in the production library.

Often, the problems that came about from this kind of error could be resolved by restoring the data of the feeder in question and placing the new program(s) into production. For example, if the Stores feeder failed because the modified program was not in production, I would restore the previous data from a back-up file and rerun the set of programs. All of the data files would be restored from the beginning point of rerunning the program(s). Changes of this nature are often called backing-off the job/program.

Chapter 4

This chapter focuses on the “nifty” features of the new Feeder system. Also, it discusses the cost/benefit analysis on the development of the new Feeder system.

Special System Features

The new Feeder system was developed to process the transaction journals for Hourly Payroll, Salary Payroll, Stores, Job Labor and Accounts Payable feeders. In constructing this system, there were several special features, which sets this system apart from other systems of its kind. The following is a list of special features that can be found in the new Feeder system:

- The conversion programs' logic is in a tabular designed file located outside the programs. This feature allows for updating the conversion logic without the need to compile the programs to make the desired enhancement(s). This innovation has limited the use of hard-code edits to only general condition edits in the conversion programs. The use of this technique is rare as it relates to the financial systems at AK Steel Corporation.
- The edit process feature uses changes made in the suspense file to update the accounting format file for the desired feeder. The suspense file consists only of transaction accounts that have not successfully passed the validation process.
- The background tracking feature tracks the status of the process which has been submitted in batch job queue. This special feature alerts the user when the batch job has been completely processed. Normally, the submitter would need to keep track of the process by using other Job Control Language (JCL) utilities; however, this feature automatically tracks the status of the process.
- The re-edit process feature allows the user to make modifications to previously validated transaction journals and to re-run the validation process on the transaction journals to insure that all of the accounts are still valid.

Special System Features

- The view report on-line feature permits the user to view the report on-line instead of having to print the report to determine the results. The feature allows the users to see results immediately after the job has been executed.
- The standardized format file feature permits the easy manipulation of data of feeders. The standardized format includes the transaction journal account, the unit amount, dollar amount, batch processing date and the original charge account. This is the standard format for all of the five feeders.

Cost/Benefit Analysis

Is the new system worth its cost? I believe it is. The new system has successfully met and exceeded the system requirements and specifications. The following is a list of the cost and benefit analysis on the new system:

Costs

- It took me approximately 874 hours to analyze, design, test, and implement the new Feeder system.
- Approximately, 5 hours was required of the end-users to test the new Feeder system.

Benefits

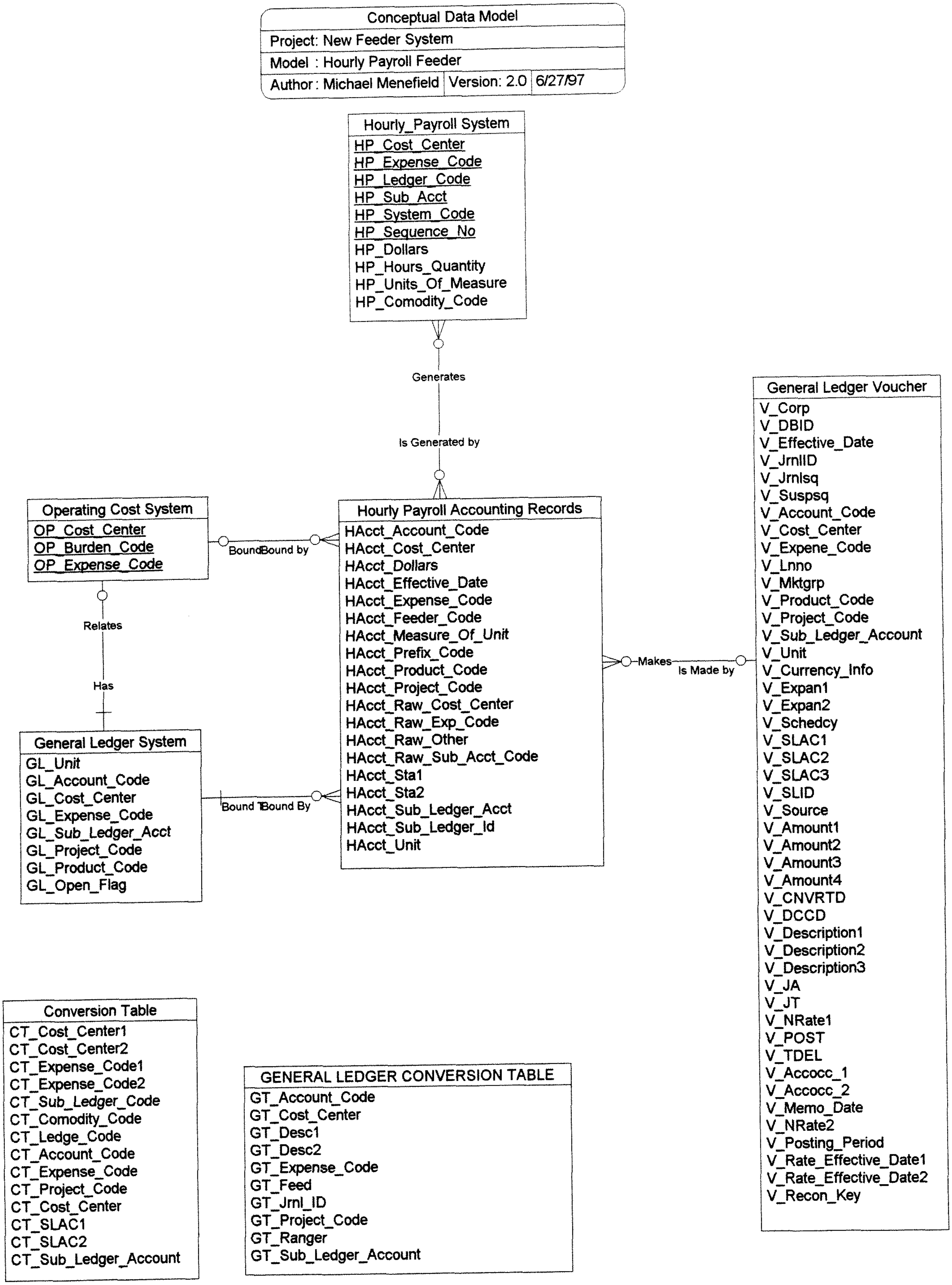
- The new Feeder system eliminated the need to re-run the General Ledger posting process because of invalid account. In addition, the General Ledger posts all five feeders' journal vouchers in three runs. The new Feeder system reduced the time to successfully post journal vouchers by approximately 1.5 to 4.0 clock hours. Also, it directly reduced the amount of time required to prepare the Operating Cost system to run the distribution process by approximately 0.5 to 2.0 clock hours.
- The new Feeder system executes each process of the feeder system in about 3.5 clock minutes. This is compared to the 30 to 90 clock minutes to execute the previous Accounts Payable processes of the former system. This change in execution time is because of the changes in process logic and the synchronization of the chart of accounts in the General Ledger and Operating Cost systems. Also, the daily processing of the feeders helped to reduce the time to execute the new Feeder system's processes.

Cost/Benefit Analysis

- The intuitive menu screens of the new Feeder system allows end-users to execute a desired process by choosing a corresponding option number. In addition, the end-users need only to enter an eight-character keyword to start the system. The previous system required the end-user to have some knowledge of how to submit batch programs in a MVS operating system.
- The new Feeder system processes daily transactions for the Stores, Job Labor and Accounts Payable feeders which reduces the amount of time needed to process the feeders at month-end. In addition, it creates daily report listings of journal transactions as well as continuously updates the monthly reports.
- The view report on-line feature of the new Feeder system reduces the amount of paper reports printed at month-end; therefore, it reduces the amount of reports in the print queue.
- The standardized format file feature permits the easy manipulation of data for all of the five feeders, which is useful in developing ad-hoc reports.

Appendix A

Appendix A contains the diagrams and other documentation used in developing the new Feeder system. In these diagrams, the shaded areas denote the aspects of the new Feeder system, which are common to the previous Feeder system.



Conceptual Data Model

Project: New Feeder System

Model : Hourly Payroll Feeder

Author: Michael Menefield | Version: 2.0 | 6/27/97

Hourly_Payroll System

- HP_Cost_Center
- HP_Expense_Code
- HP_Ledger_Code
- HP_Sub_Acct
- HP_System_Code
- HP_Sequence_No
- HP_Dollars
- HP_Hours_Quantity
- HP_Units_Of_Measure
- HP_Comodity_Code

Operating Cost System

- OP_Cost_Center
- OP_Burden_Code
- OP_Expense_Code

Hourly Payroll Accounting Records

- HAcct_Account_Code
- HAcct_Cost_Center
- HAcct_Dollars
- HAcct_Effective_Date
- HAcct_Expense_Code
- HAcct_Feeder_Code
- HAcct_Measure_Of_Unit
- HAcct_Prefix_Code
- HAcct_Product_Code
- HAcct_Project_Code
- HAcct_Raw_Cost_Center
- HAcct_Raw_Exp_Code
- HAcct_Raw_Other
- HAcct_Raw_Sub_Acct_Code
- HAcct_Sta1
- HAcct_Sta2
- HAcct_Sub_Ledger_Acct
- HAcct_Sub_Ledger_Id
- HAcct_Unit

General Ledger System

- GL_Unit
- GL_Account_Code
- GL_Cost_Center
- GL_Expense_Code
- GL_Sub_Ledger_Acct
- GL_Project_Code
- GL_Product_Code
- GL_Open_Flag

General Ledger Voucher

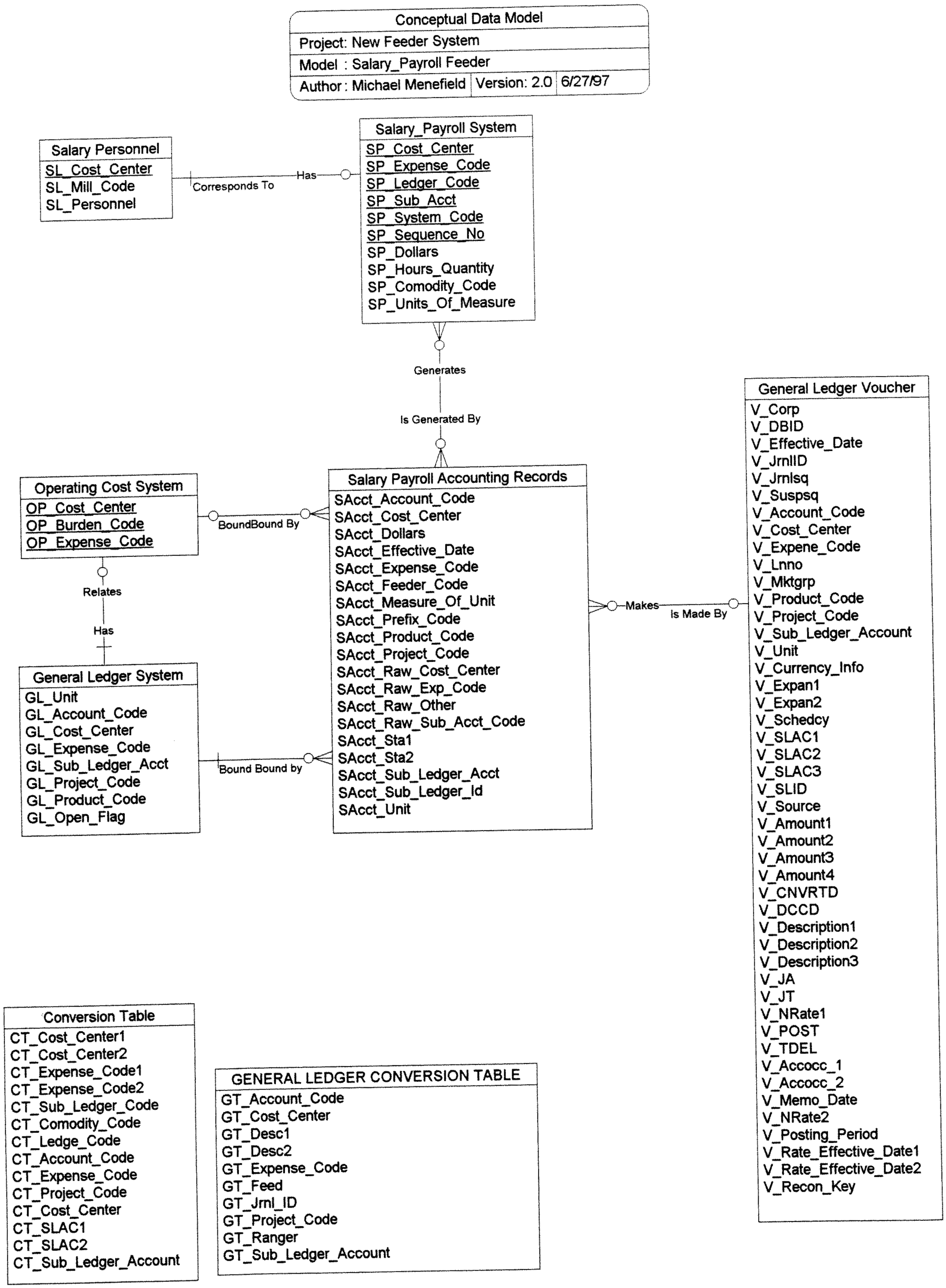
- V_Corp
- V_DBID
- V_Effective_Date
- V_JrnlID
- V_Jrnlsq
- V_Suspsq
- V_Account_Code
- V_Cost_Center
- V_Expene_Code
- V_Lnno
- V_Mktgrp
- V_Product_Code
- V_Project_Code
- V_Sub_Ledger_Account
- V_Unit
- V_Currency_Info
- V_Expan1
- V_Expan2
- V_Schedcy
- V_SLAC1
- V_SLAC2
- V_SLAC3
- V_SLID
- V_Source
- V_Amount1
- V_Amount2
- V_Amount3
- V_Amount4
- V_CNVRTD
- V_DCCD
- V_Description1
- V_Description2
- V_Description3
- V_JA
- V_JT
- V_NRate1
- V_POST
- V_TDEL
- V_Accocc_1
- V_Accocc_2
- V_Memo_Date
- V_NRate2
- V_Posting_Period
- V_Rate_Effective_Date1
- V_Rate_Effective_Date2
- V_Recon_Key

Conversion Table

- CT_Cost_Center1
- CT_Cost_Center2
- CT_Expense_Code1
- CT_Expense_Code2
- CT_Sub_Ledger_Code
- CT_Comodity_Code
- CT_Ledge_Code
- CT_Account_Code
- CT_Expense_Code
- CT_Project_Code
- CT_Cost_Center
- CT_SLAC1
- CT_SLAC2
- CT_Sub_Ledger_Account

GENERAL LEDGER CONVERSION TABLE

- GT_Account_Code
- GT_Cost_Center
- GT_Desc1
- GT_Desc2
- GT_Expense_Code
- GT_Feed
- GT_Jrnl_ID
- GT_Project_Code
- GT_Ranger
- GT_Sub_Ledger_Account



Conceptual Data Model		
Project: New Feeder System		
Model : Salary_Payroll Feeder		
Author : Michael Menefield	Version: 2.0	6/27/97

Salary Personnel
- SL_Cost_Center
- SL_Mill_Code
- SL_Personnel

Salary_Payroll System
- SP_Cost_Center
- SP_Expense_Code
- SP_Ledger_Code
- SP_Sub_Acct
- SP_System_Code
- SP_Sequence_No
- SP_Dollars
- SP_Hours_Quantity
- SP_Comodity_Code
- SP_Units_Of_Measure

Operating Cost System
- OP_Cost_Center
- OP_Burden_Code
- OP_Expense_Code

Salary Payroll Accounting Records
- SAcct_Account_Code
- SAcct_Cost_Center
- SAcct_Dollars
- SAcct_Effective_Date
- SAcct_Expense_Code
- SAcct_Feeder_Code
- SAcct_Measure_Of_Unit
- SAcct_Prefix_Code
- SAcct_Product_Code
- SAcct_Project_Code
- SAcct_Raw_Cost_Center
- SAcct_Raw_Exp_Code
- SAcct_Raw_Other
- SAcct_Raw_Sub_Acct_Code
- SAcct_Sta1
- SAcct_Sta2
- SAcct_Sub_Ledger_Acct
- SAcct_Sub_Ledger_Id
- SAcct_Unit

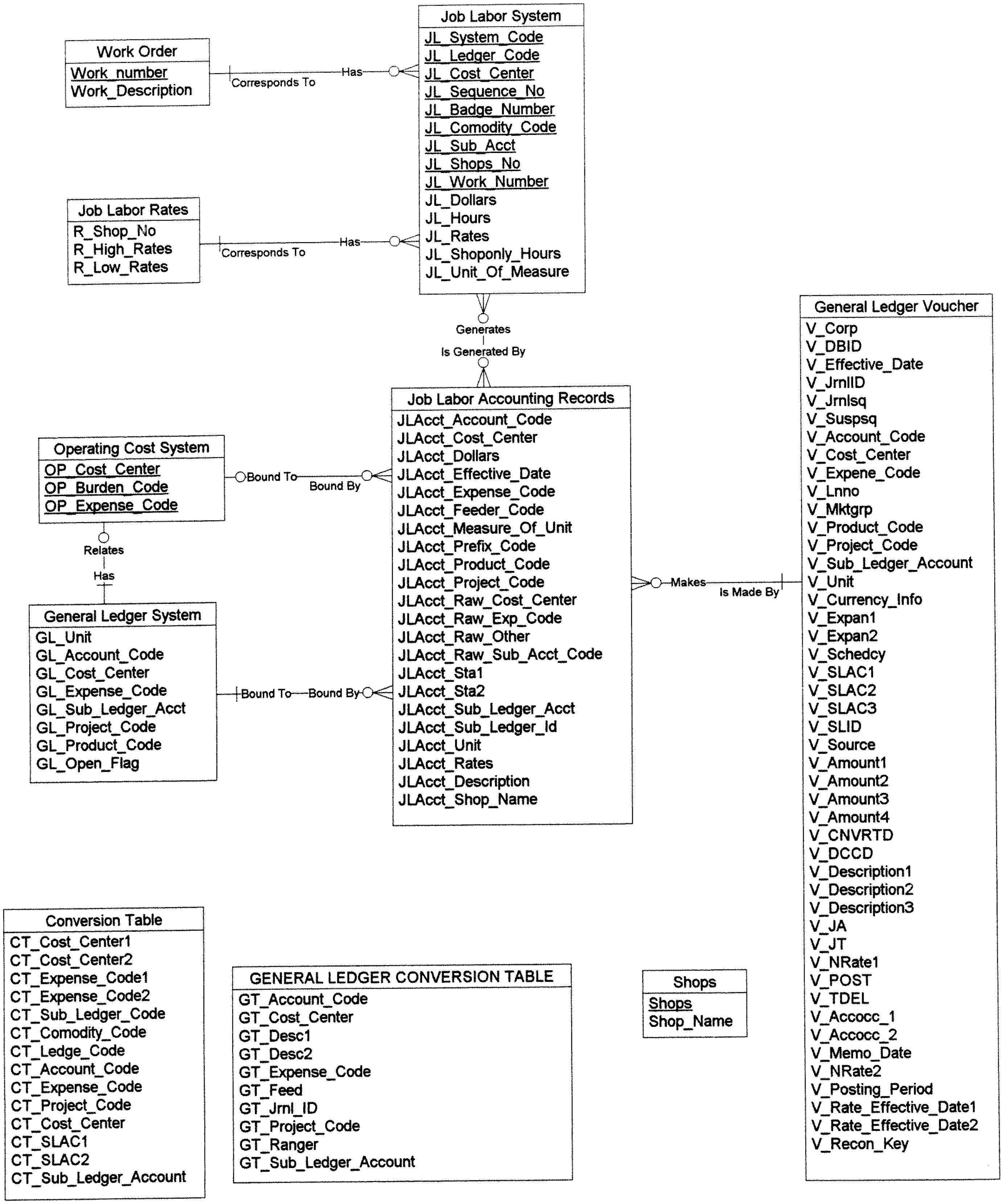
General Ledger System
- GL_Unit
- GL_Account_Code
- GL_Cost_Center
- GL_Expense_Code
- GL_Sub_Ledger_Acct
- GL_Project_Code
- GL_Product_Code
- GL_Open_Flag

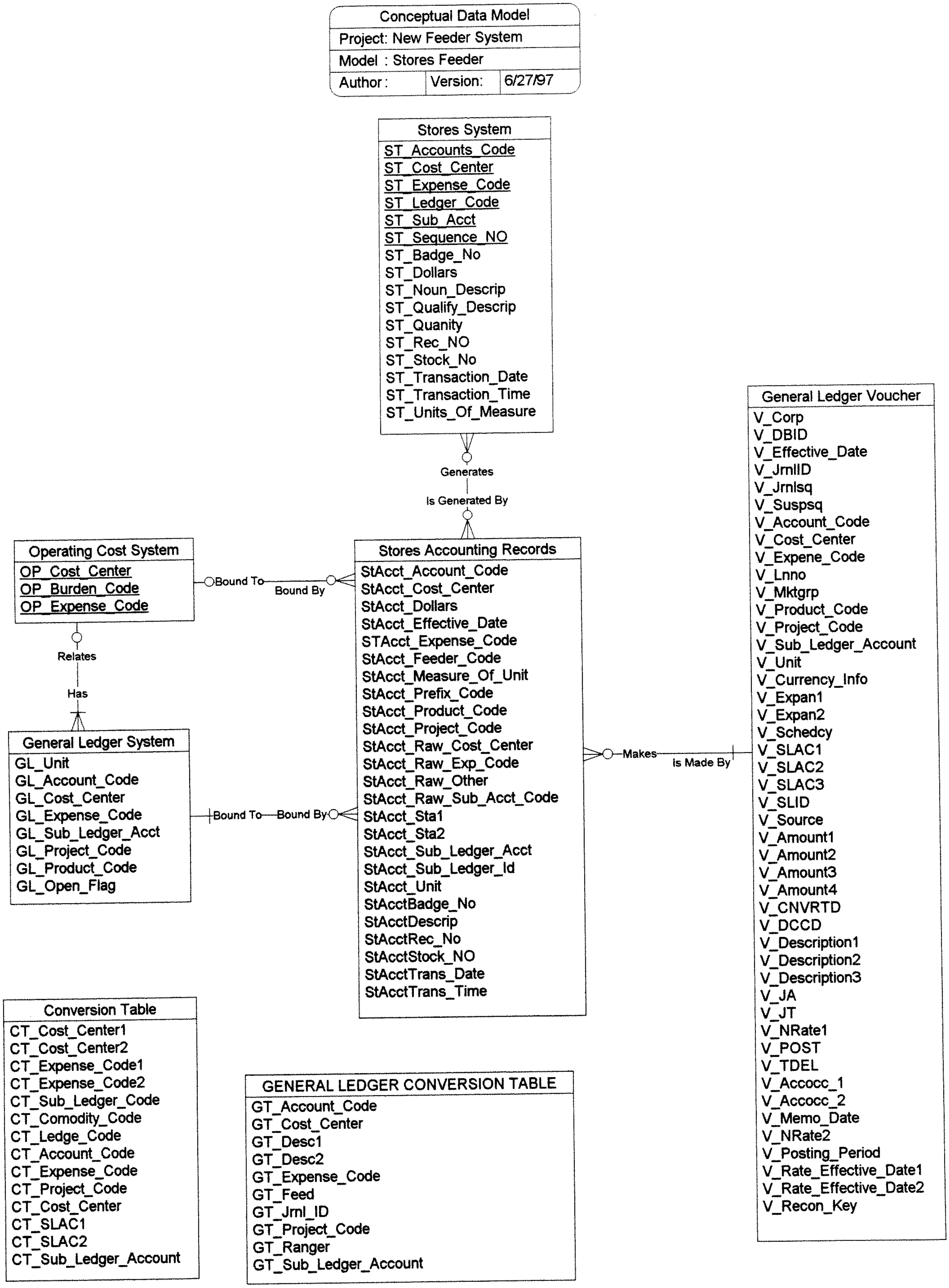
General Ledger Voucher
- V_Corp
- V_DBID
- V_Effective_Date
- V_JrnlID
- V_Jrnlsq
- V_Suspsq
- V_Account_Code
- V_Cost_Center
- V_Expene_Code
- V_Lnno
- V_Mktgrp
- V_Product_Code
- V_Project_Code
- V_Sub_Ledger_Account
- V_Unit
- V_Currency_Info
- V_Expan1
- V_Expan2
- V_Schedcy
- V_SLAC1
- V_SLAC2
- V_SLAC3
- V_SLID
- V_Source
- V_Amount1
- V_Amount2
- V_Amount3
- V_Amount4
- V_CNVRTD
- V_DCCD
- V_Description1
- V_Description2
- V_Description3
- V_JA
- V_JT
- V_NRate1
- V_POST
- V_TDEL
- V_Accocc_1
- V_Accocc_2
- V_Memo_Date
- V_NRate2
- V_Posting_Period
- V_Rate_Effective_Date1
- V_Rate_Effective_Date2
- V_Recon_Key

Conversion Table
- CT_Cost_Center1
- CT_Cost_Center2
- CT_Expense_Code1
- CT_Expense_Code2
- CT_Sub_Ledger_Code
- CT_Comodity_Code
- CT_Ledge_Code
- CT_Account_Code
- CT_Expense_Code
- CT_Project_Code
- CT_Cost_Center
- CT_SLAC1
- CT_SLAC2
- CT_Sub_Ledger_Account

GENERAL LEDGER CONVERSION TABLE
- GT_Account_Code
- GT_Cost_Center
- GT_Desc1
- GT_Desc2
- GT_Expense_Code
- GT_Feed
- GT_Jrnl_ID
- GT_Project_Code
- GT_Ranger
- GT_Sub_Ledger_Account

Conceptual Data Model		
Project: New Feeder Sysem		
Model : Job Labor Feeder		
Author : Michael Menefield	Version: 2.0	6/27/97





Conceptual Data Model
Project: New Feeder System
Model : Stores Feeder
Author: | Version: | 6/27/97

Stores System
ST_Accounts_Code
ST_Cost_Center
ST_Expense_Code
ST_Ledger_Code
ST_Sub_Acct
ST_Sequence_NO
ST_Badge_No
ST_Dollars
ST_Noun_Descrip
ST_Qualify_Descrip
ST_Quanity
ST_Rec_NO
ST_Stock_No
ST_Transaction_Date
ST_Transaction_Time
ST_Units_Of_Measure

Operating Cost System
OP_Cost_Center
OP_Burden_Code
OP_Expense_Code

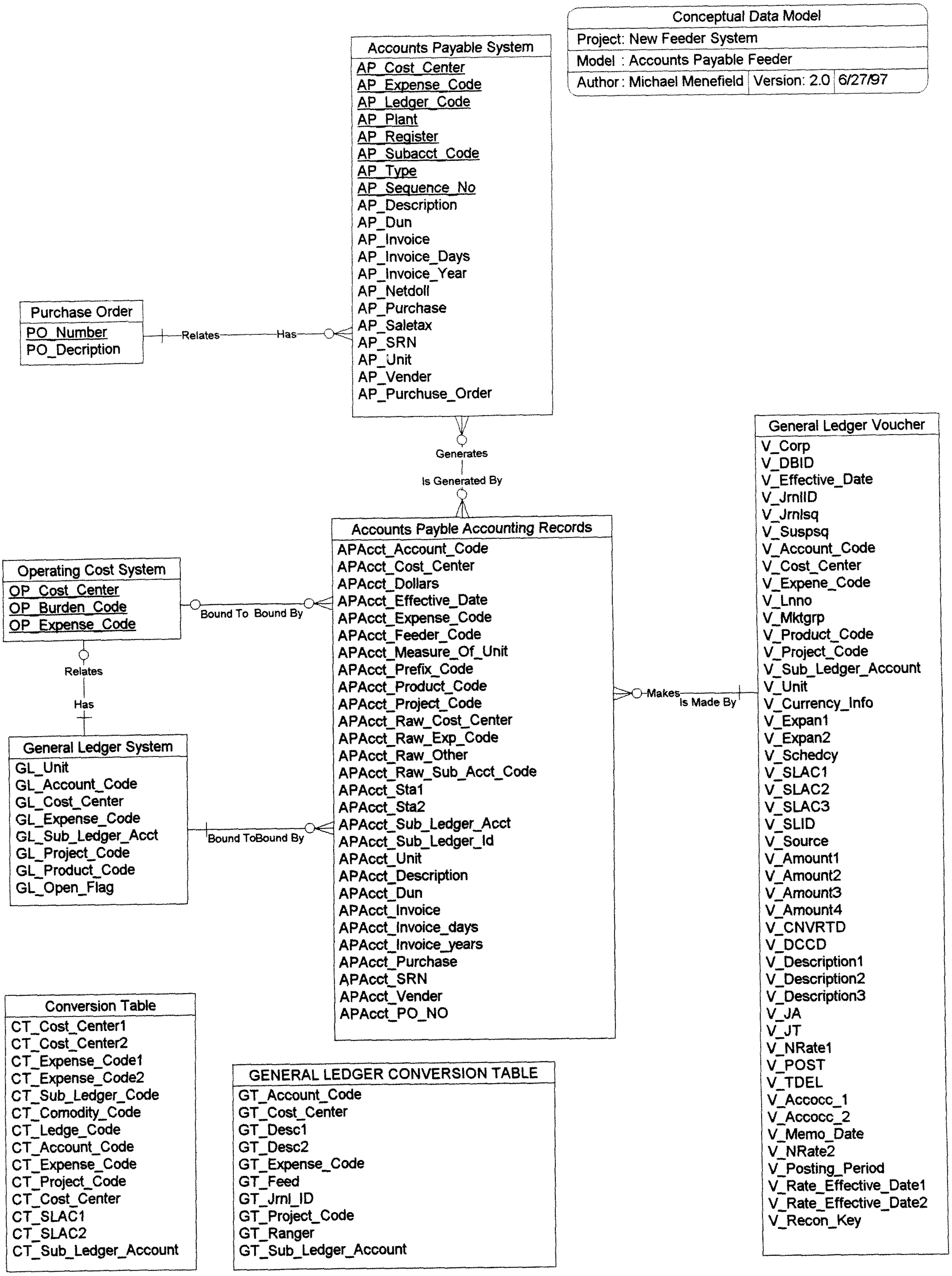
Stores Accounting Records
StAcct_Account_Code
StAcct_Cost_Center
StAcct_Dollars
StAcct_Effective_Date
STAcct_Expense_Code
StAcct_Feeder_Code
StAcct_Measure_Of_Unit
StAcct_Prefix_Code
StAcct_Product_Code
StAcct_Project_Code
StAcct_Raw_Cost_Center
StAcct_Raw_Exp_Code
StAcct_Raw_Other
StAcct_Raw_Sub_Acct_Code
StAcct_Sta1
StAcct_Sta2
StAcct_Sub_Ledger_Acct
StAcct_Sub_Ledger_Id
StAcct_Unit
StAcctBadge_No
StAcctDescrip
StAcctRec_No
StAcctStock_NO
StAcctTrans_Date
StAcctTrans_Time

General Ledger System
GL_Unit
GL_Account_Code
GL_Cost_Center
GL_Expense_Code
GL_Sub_Ledger_Acct
GL_Project_Code
GL_Product_Code
GL_Open_Flag

General Ledger Voucher
V_Corp
V_DBID
V_Effective_Date
V_JrnlID
V_Jrnlsq
V_Suspsq
V_Account_Code
V_Cost_Center
V_Expene_Code
V_Lnno
V_Mktgrp
V_Product_Code
V_Project_Code
V_Sub_Ledger_Account
V_Unit
V_Currency_Info
V_Expan1
V_Expan2
V_Schedcy
V_SLAC1
V_SLAC2
V_SLAC3
V_SLID
V_Source
V_Amount1
V_Amount2
V_Amount3
V_Amount4
V_CNVRTD
V_DCCD
V_Description1
V_Description2
V_Description3
V_JA
V_JT
V_NRate1
V_POST
V_TDEL
V_Accocc_1
V_Accocc_2
V_Memo_Date
V_NRate2
V_Posting_Period
V_Rate_Effective_Date1
V_Rate_Effective_Date2
V_Recon_Key

Conversion Table
CT_Cost_Center1
CT_Cost_Center2
CT_Expense_Code1
CT_Expense_Code2
CT_Sub_Ledger_Code
CT_Comodity_Code
CT_Ledge_Code
CT_Account_Code
CT_Expense_Code
CT_Project_Code
CT_Cost_Center
CT_SLAC1
CT_SLAC2
CT_Sub_Ledger_Account

GENERAL LEDGER CONVERSION TABLE
GT_Account_Code
GT_Cost_Center
GT_Desc1
GT_Desc2
GT_Expense_Code
GT_Feed
GT_Jrnl_ID
GT_Project_Code
GT_Ranger
GT_Sub_Ledger_Account



Conceptual Data Model		
Project: New Feeder System		
Model : Accounts Payable Feeder		
Author : Michael Menefield	Version: 2.0	6/27/97

Accounts Payable System

AP_Cost_Center
AP_Expense_Code
AP_Ledger_Code
AP_Plant
AP_Register
AP_Subacct_Code
AP_Type
AP_Sequence_No
AP_Description
AP_Dun
AP_Invoice
AP_Invoice_Days
AP_Invoice_Year
AP_Netdoll
AP_Purchase
AP_Saletax
AP_SRN
AP_Unit
AP_Vender
AP_Purchuse_Order

Purchase Order

PO_Number
PO_Decription

Accounts Payble Accounting Records

APAcct_Account_Code
APAcct_Cost_Center
APAcct_Dollars
APAcct_Effective_Date
APAcct_Expense_Code
APAcct_Feeder_Code
APAcct_Measure_Of_Unit
APAcct_Prefix_Code
APAcct_Product_Code
APAcct_Project_Code
APAcct_Raw_Cost_Center
APAcct_Raw_Exp_Code
APAcct_Raw_Other
APAcct_Raw_Sub_Acct_Code
APAcct_Sta1
APAcct_Sta2
APAcct_Sub_Ledger_Acct
APAcct_Sub_Ledger_Id
APAcct_Unit
APAcct_Description
APAcct_Dun
APAcct_Invoice
APAcct_Invoice_days
APAcct_Invoice_years
APAcct_Purchase
APAcct_SRN
APAcct_Vender
APAcct_PO_NO

Operating Cost System

OP_Cost_Center
OP_Burden_Code
OP_Expense_Code

General Ledger System

GL_Unit
GL_Account_Code
GL_Cost_Center
GL_Expense_Code
GL_Sub_Ledger_Acct
GL_Project_Code
GL_Product_Code
GL_Open_Flag

General Ledger Voucher

V_Corp
V_DBID
V_Effective_Date
V_JrnlID
V_Jrnlsq
V_Suspsq
V_Account_Code
V_Cost_Center
V_Expene_Code
V_Lnno
V_Mktgrp
V_Product_Code
V_Project_Code
V_Sub_Ledger_Account
V_Unit
V_Currency_Info
V_Expan1
V_Expan2
V_Schedcy
V_SLAC1
V_SLAC2
V_SLAC3
V_SLID
V_Source
V_Amount1
V_Amount2
V_Amount3
V_Amount4
V_CNVRTD
V_DCCD
V_Description1
V_Description2
V_Description3
V_JA
V_JT
V_NRate1
V_POST
V_TDEL
V_Accocc_1
V_Accocc_2
V_Memo_Date
V_NRate2
V_Posting_Period
V_Rate_Effective_Date1
V_Rate_Effective_Date2
V_Recon_Key

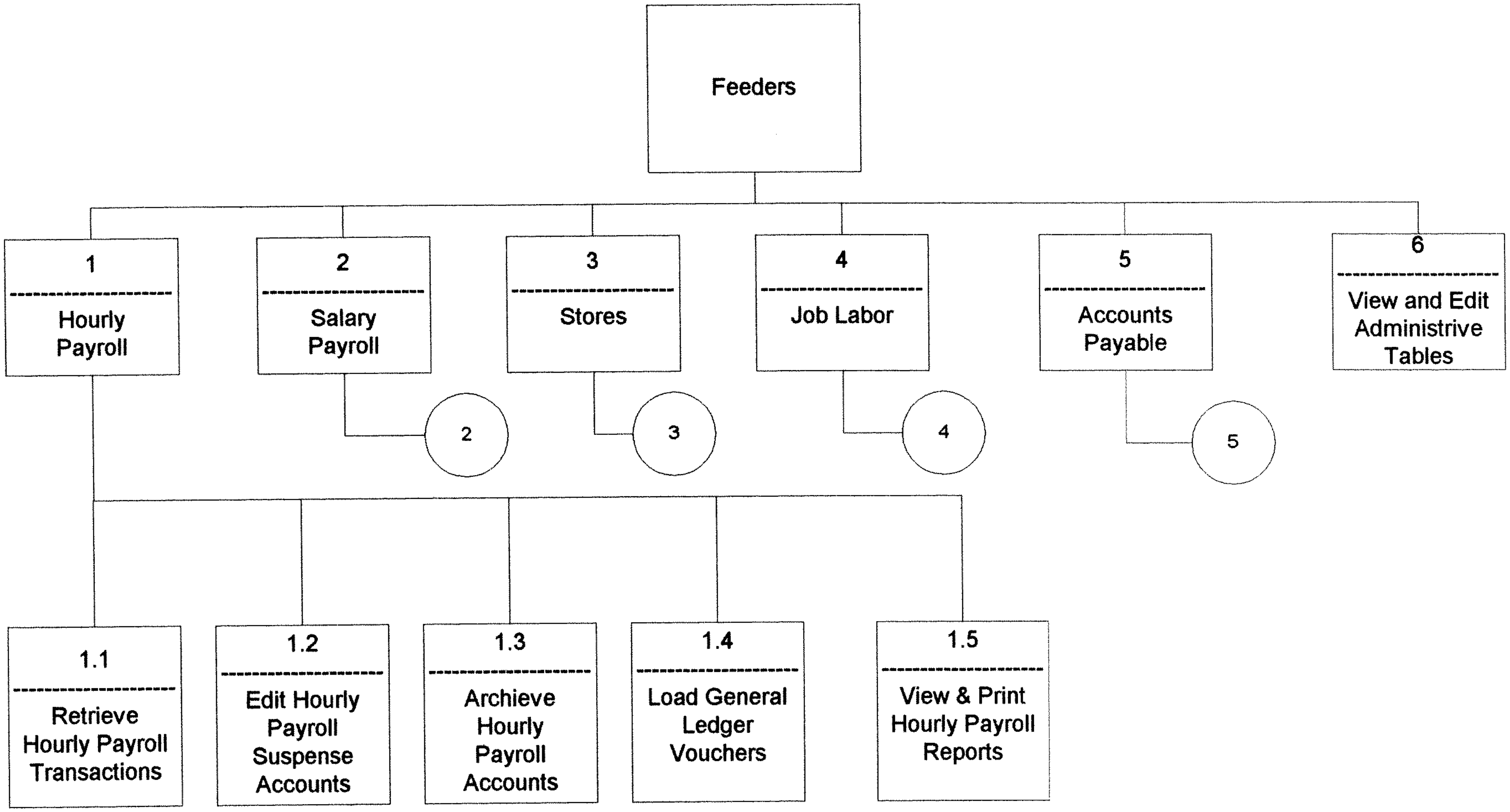
Conversion Table

CT_Cost_Center1
CT_Cost_Center2
CT_Expense_Code1
CT_Expense_Code2
CT_Sub_Ledger_Code
CT_Comodity_Code
CT_Ledge_Code
CT_Account_Code
CT_Expense_Code
CT_Project_Code
CT_Cost_Center
CT_SLAC1
CT_SLAC2
CT_Sub_Ledger_Account

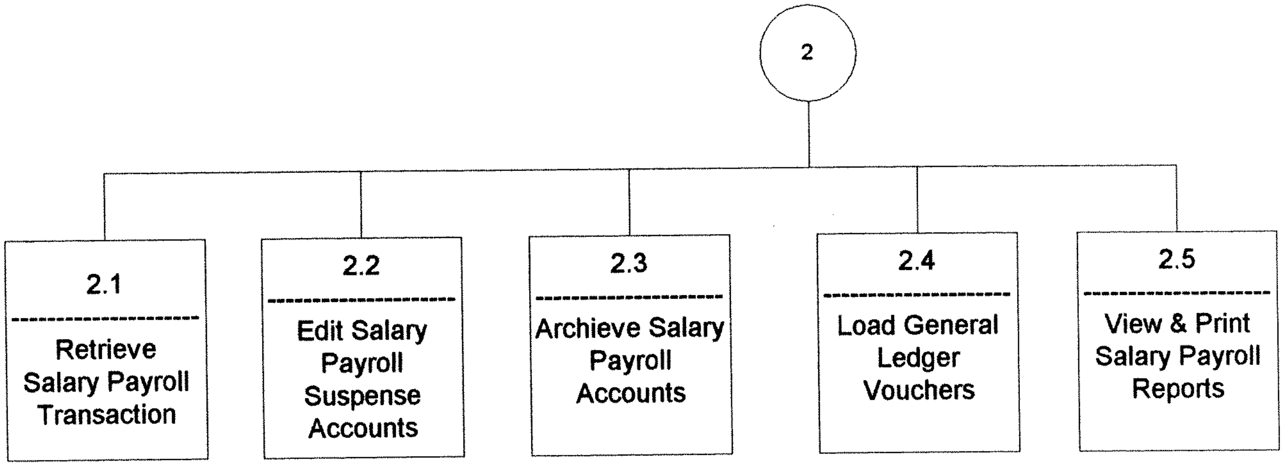
GENERAL LEDGER CONVERSION TABLE

GT_Account_Code
GT_Cost_Center
GT_Desc1
GT_Desc2
GT_Expense_Code
GT_Feed
GT_Jrnl_ID
GT_Project_Code
GT_Ranger
GT_Sub_Ledger_Account

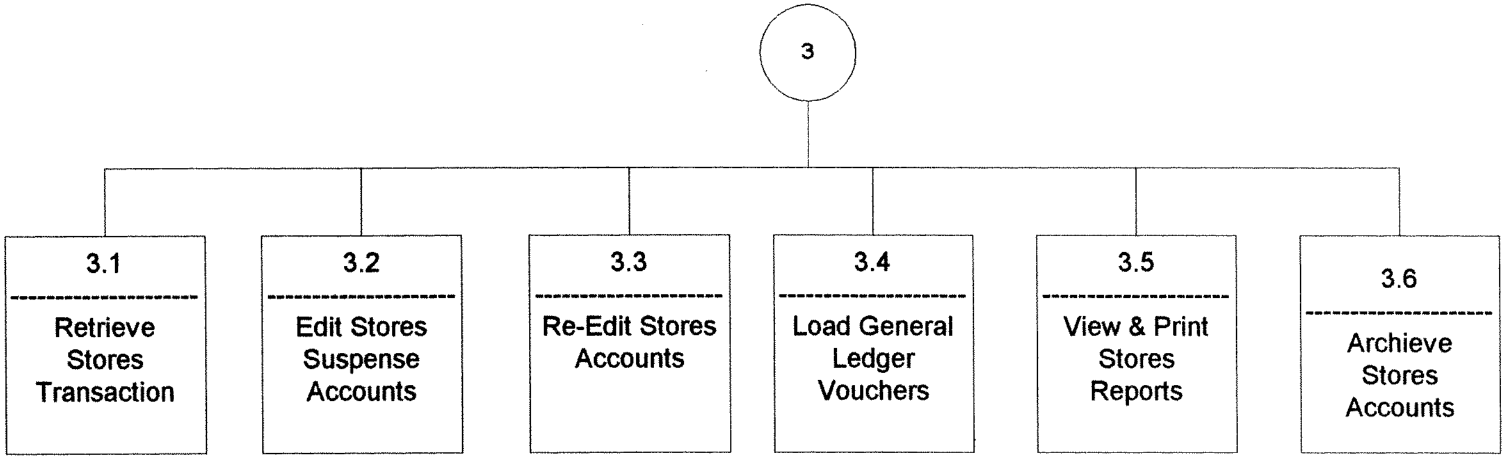
New Feeder System
Decomposition Diagram


Feeders
1
Hourly Payroll
2
Salary Payroll
3
Stores
4
Job Labor
5
Accounts Payable
6
View and Edit Administrive Tables
2
3
4
5
1.1
Retrieve Hourly Payroll Transactions
1.2
Edit Hourly Payroll Suspense Accounts
1.3
Archieve Hourly Payroll Accounts
1.4
Load General Ledger Vouchers
1.5
View & Print Hourly Payroll Reports

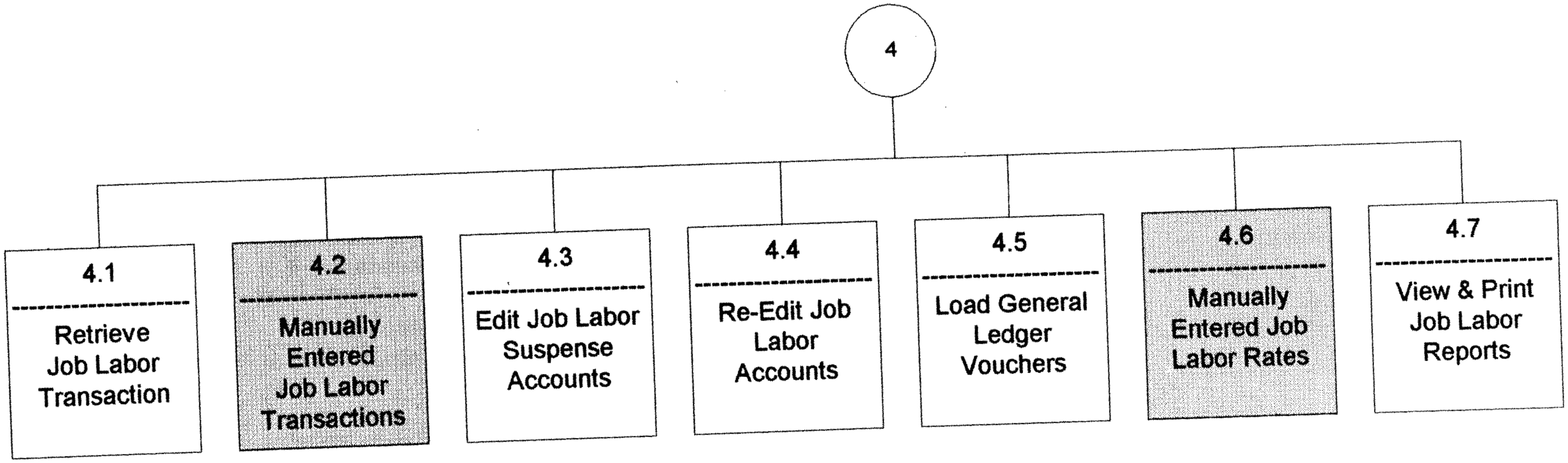
New Feeder System
Decomposition Diagram


2
2.1
Retrieve Salary Payroll Transaction
2.2
Edit Salary Payroll Suspense Accounts
2.3
Archieve Salary Payroll Accounts
2.4
Load General Ledger Vouchers
2.5
View & Print Salary Payroll Reports

New Feeder System
Decomposition Diagram

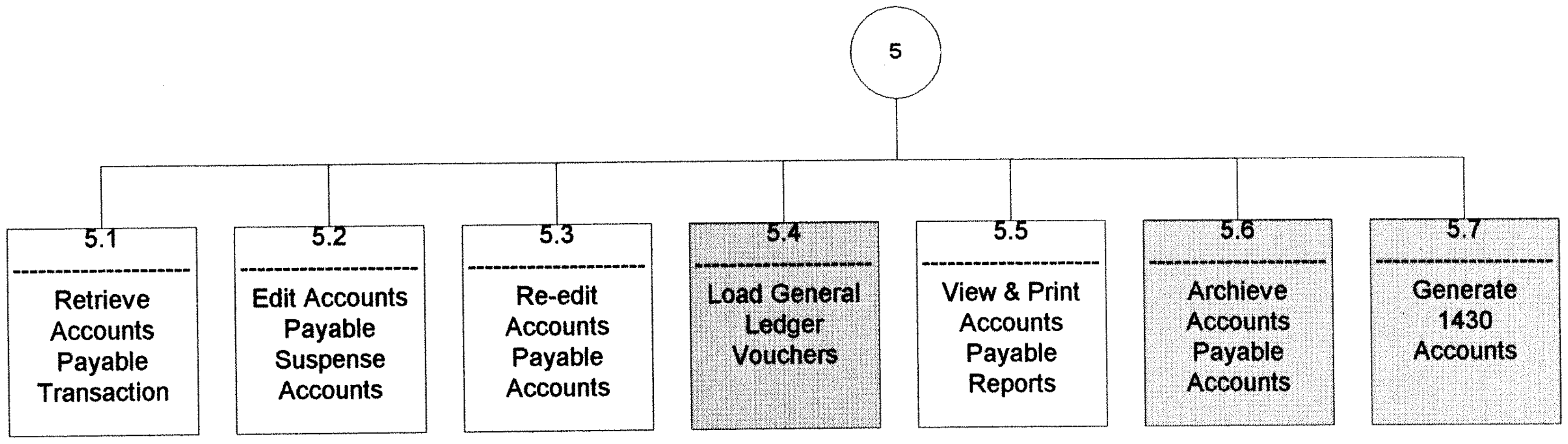

3
3.1
Retrieve Stores Transaction
3.2
Edit Stores Suspense Accounts
3.3
Re-Edit Stores Accounts
3.4
Load General Ledger Vouchers
3.5
View & Print Stores Reports
3.6
Archieve Stores Accounts

New Feeder System
Decomposition Diagram

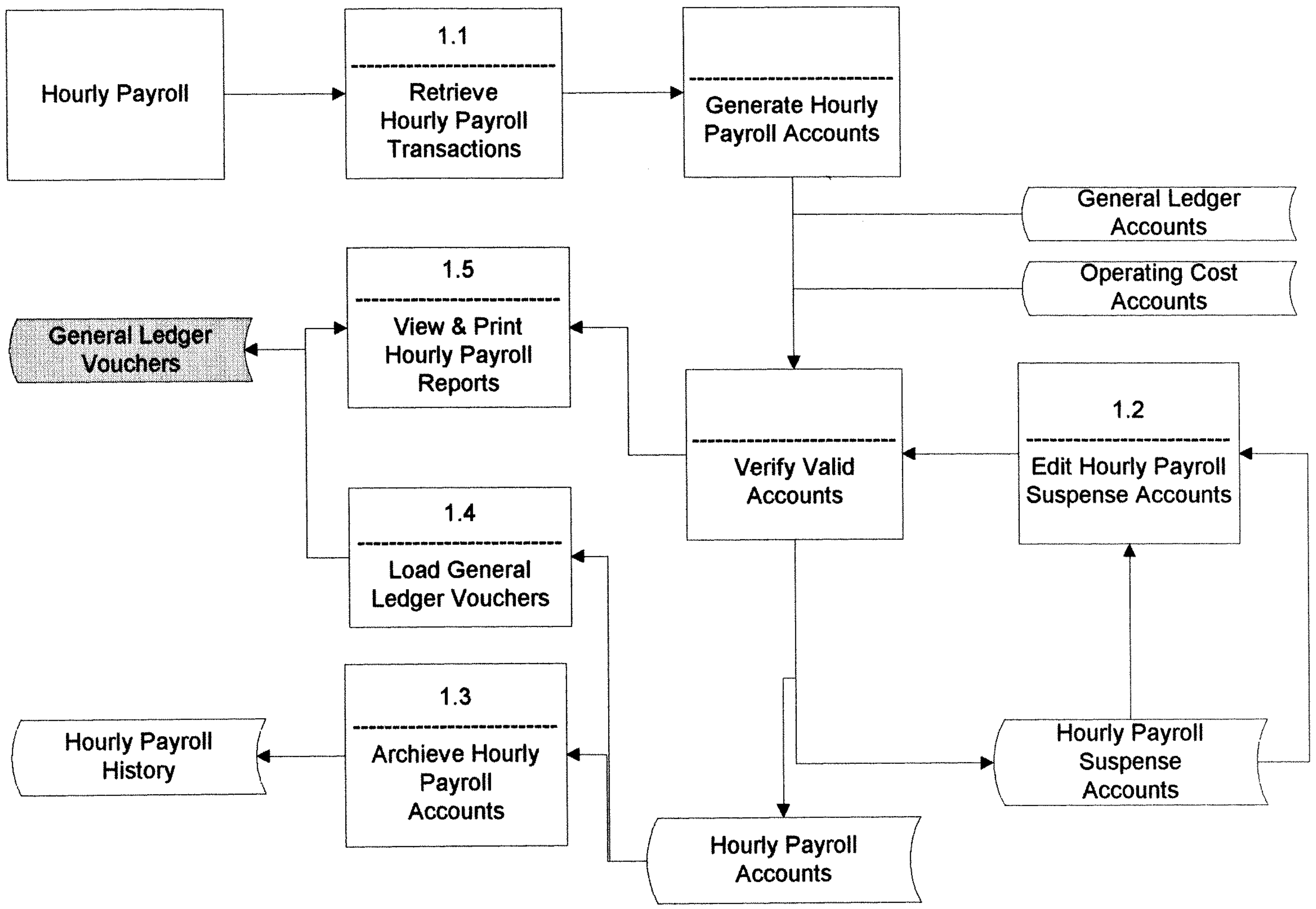


4
4.1
Retrieve Job Labor Transaction
4.2
Manually Entered Job Labor Transactions
4.3
Edit Job Labor Suspense Accounts
4.4
Re-Edit Job Labor Accounts
4.5
Load General Ledger Vouchers
4.6
Manually Entered Job Labor Rates
4.7
View & Print Job Labor Reports

New Feeder System
Decomposition Diagram

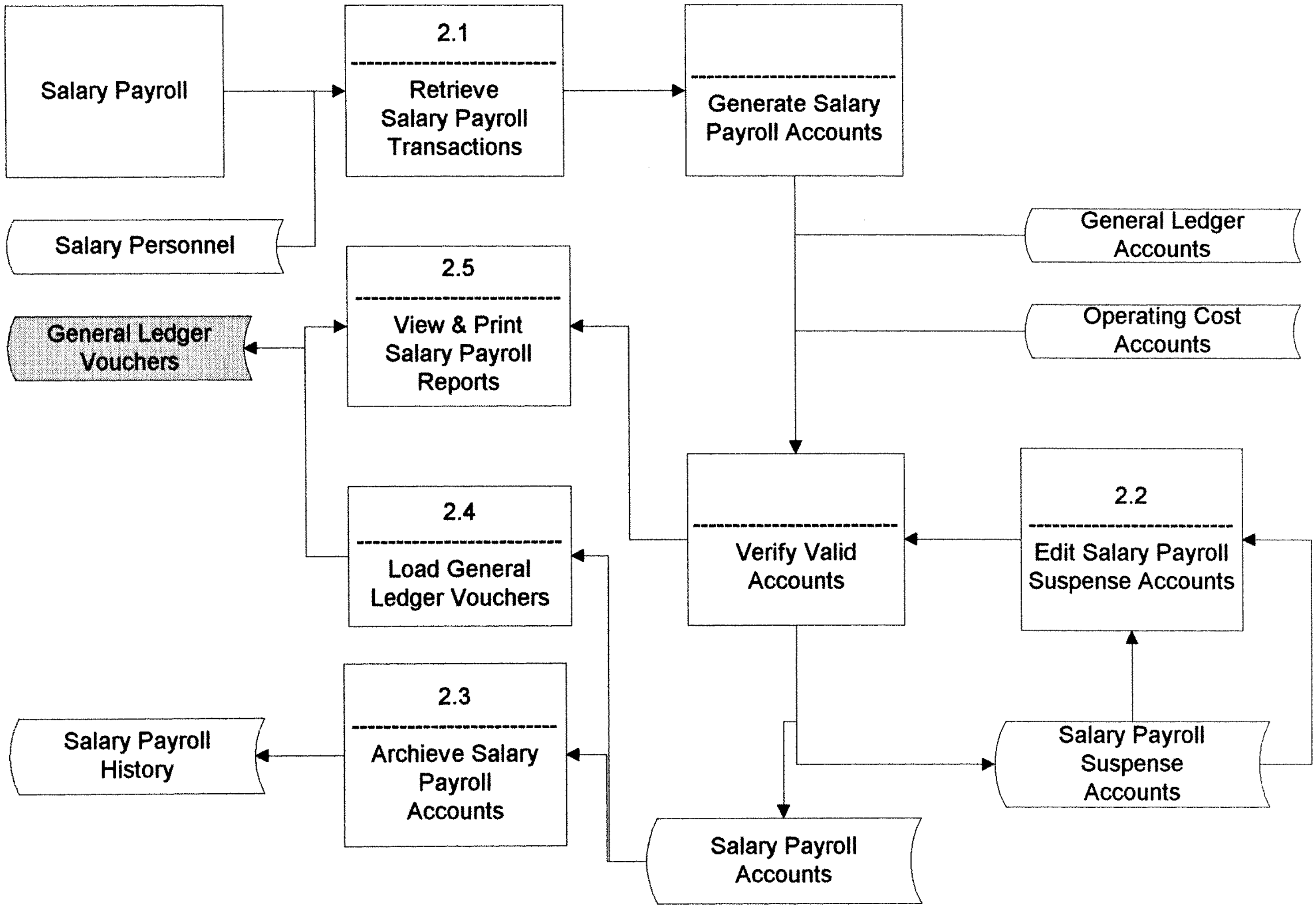

5
5.1
Retrieve Accounts Payable Transaction
5.2
Edit Accounts Payable Suspense Accounts
5.3
Re-edit Accounts Payable Accounts
5.4
Load General Ledger Vouchers
5.5
View & Print Accounts Payable Reports
5.6
Archieve Accounts Payable Accounts
5.7
Generate 1430 Accounts

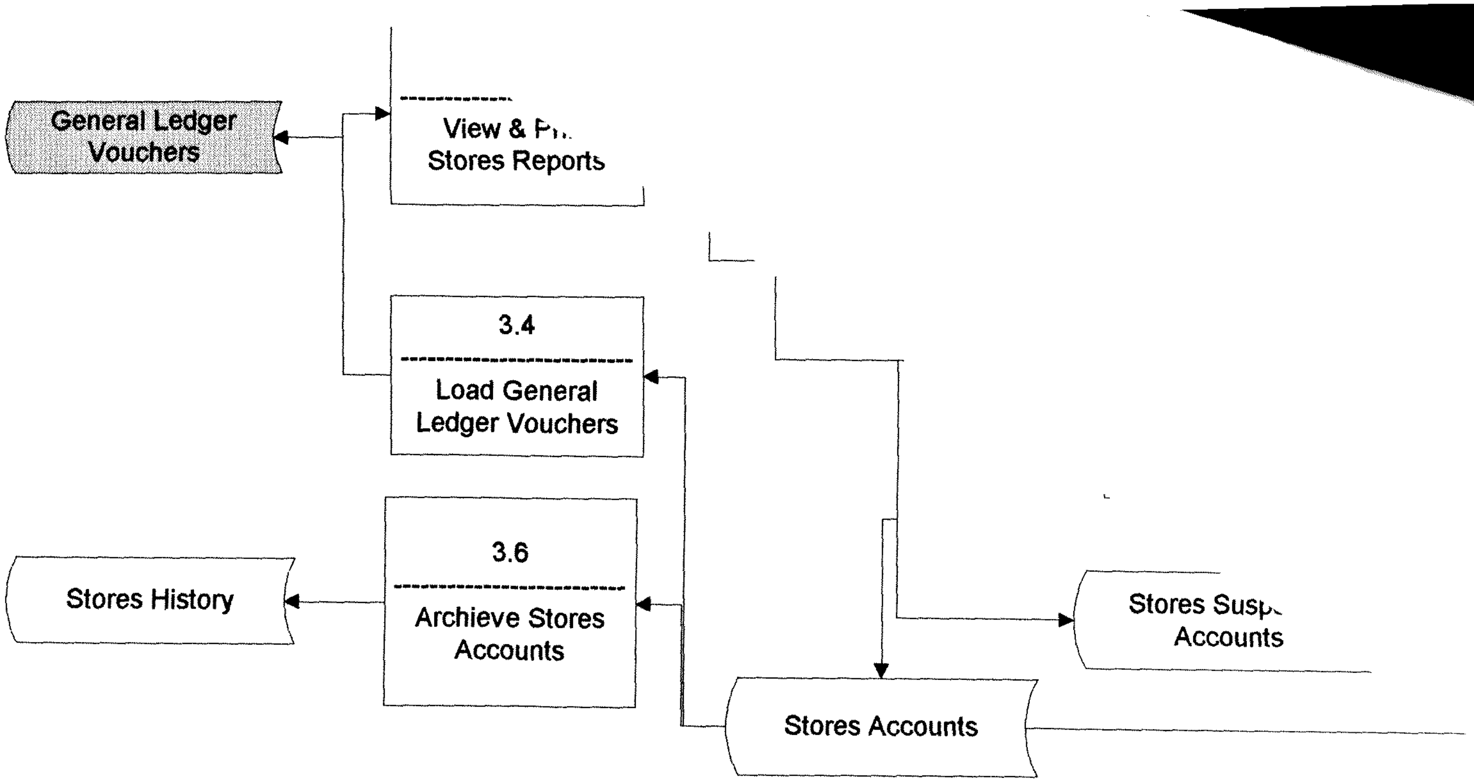
Hourly Payroll Process Model



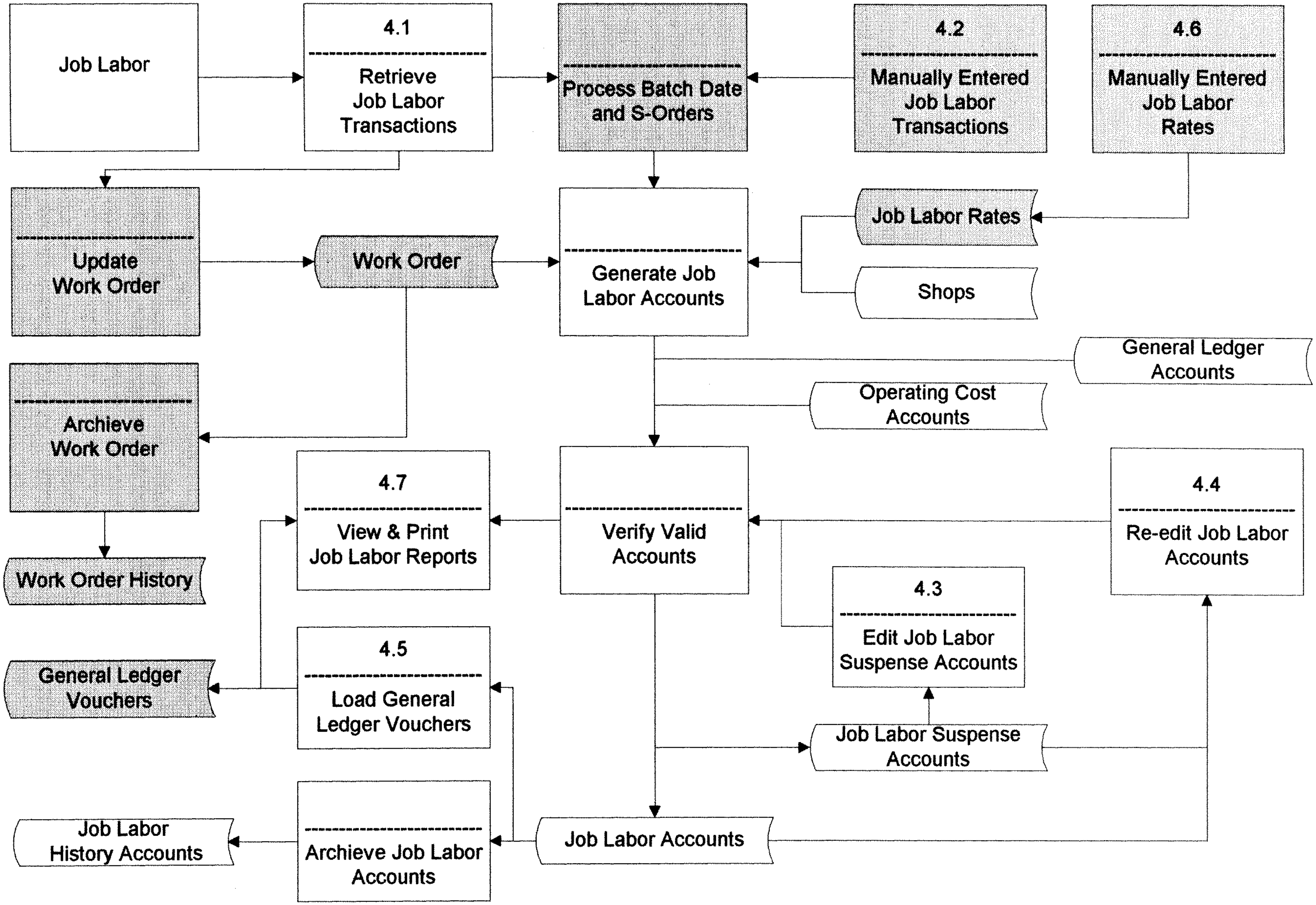
Hourly Payroll
1.1
Retrieve Hourly Payroll Transactions
Generate Hourly Payroll Accounts
General Ledger Accounts
Operating Cost Accounts
1.5
View & Print Hourly Payroll Reports
General Ledger Vouchers
Verify Valid Accounts
1.2
Edit Hourly Payroll Suspense Accounts
1.4
Load General Ledger Vouchers
1.3
Archieve Hourly Payroll Accounts
Hourly Payroll History
Hourly Payroll Suspense Accounts
Hourly Payroll Accounts

Salary Payroll Process Model

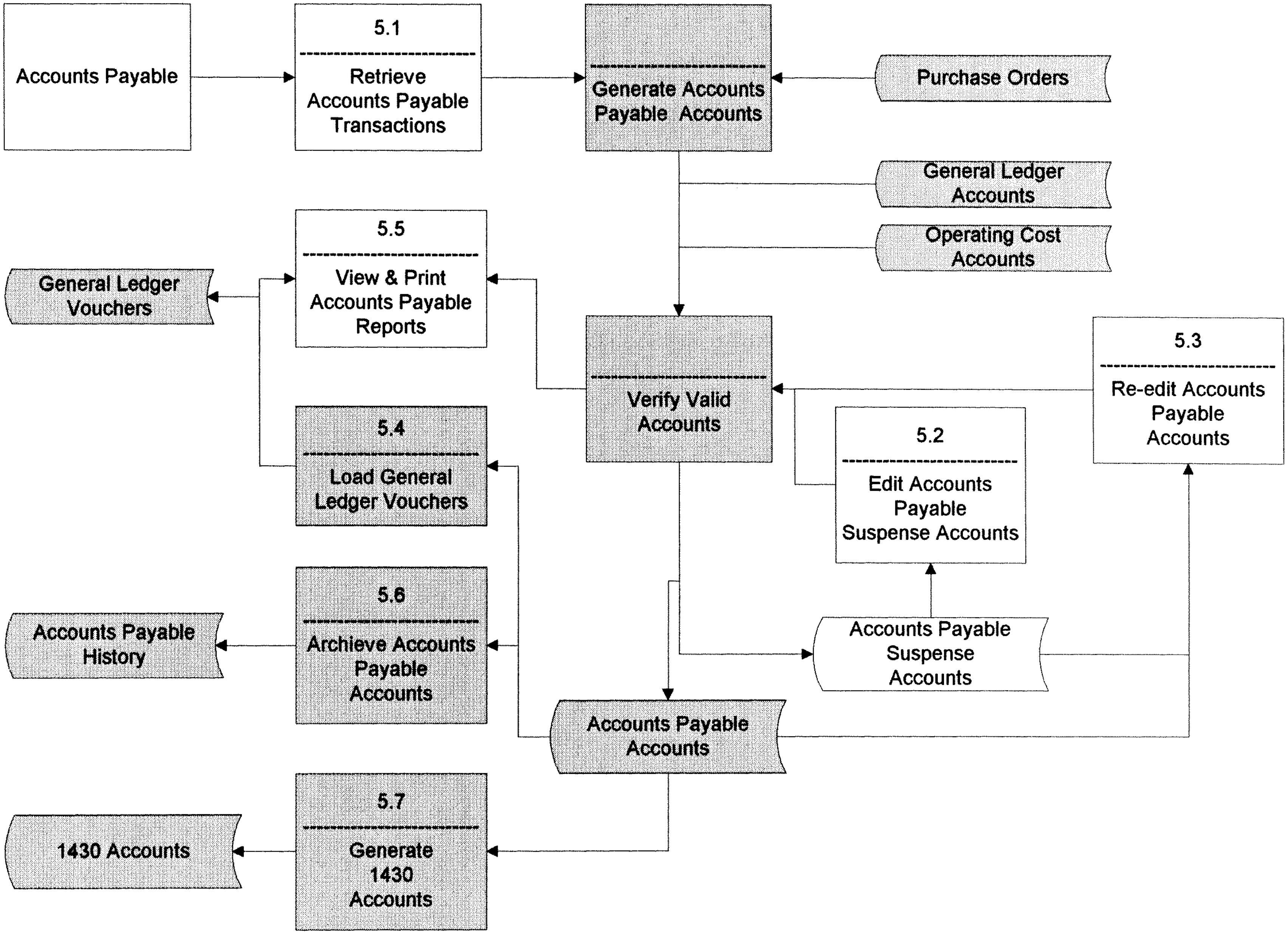

Salary Payroll
2.1
Retrieve Salary Payroll Transactions
Generate Salary Payroll Accounts
Salary Personnel
General Ledger Accounts
2.5
View & Print Salary Payroll Reports
General Ledger Vouchers
Operating Cost Accounts
Verify Valid Accounts
2.2
Edit Salary Payroll Suspense Accounts
2.4
Load General Ledger Vouchers
2.3
Archieve Salary Payroll Accounts
Salary Payroll History
Salary Payroll Suspense Accounts
Salary Payroll Accounts

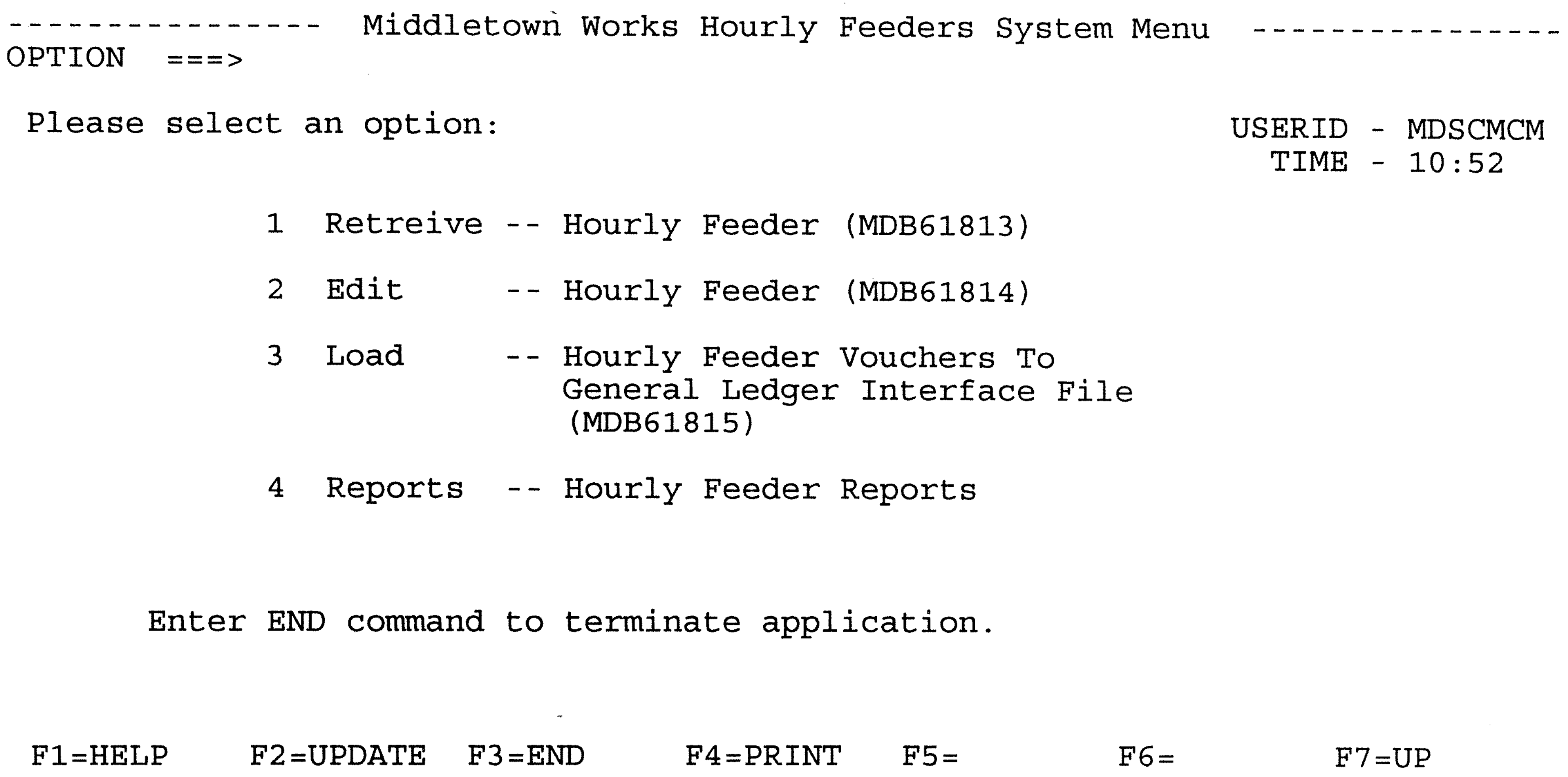

General Ledger Vouchers
View & Pr
Stores Reports
3.4
Load General Ledger Vouchers
3.6
Archieve Stores Accounts
Stores History
Stores Susp
Accounts
Stores Accounts

Job Labor Process Model

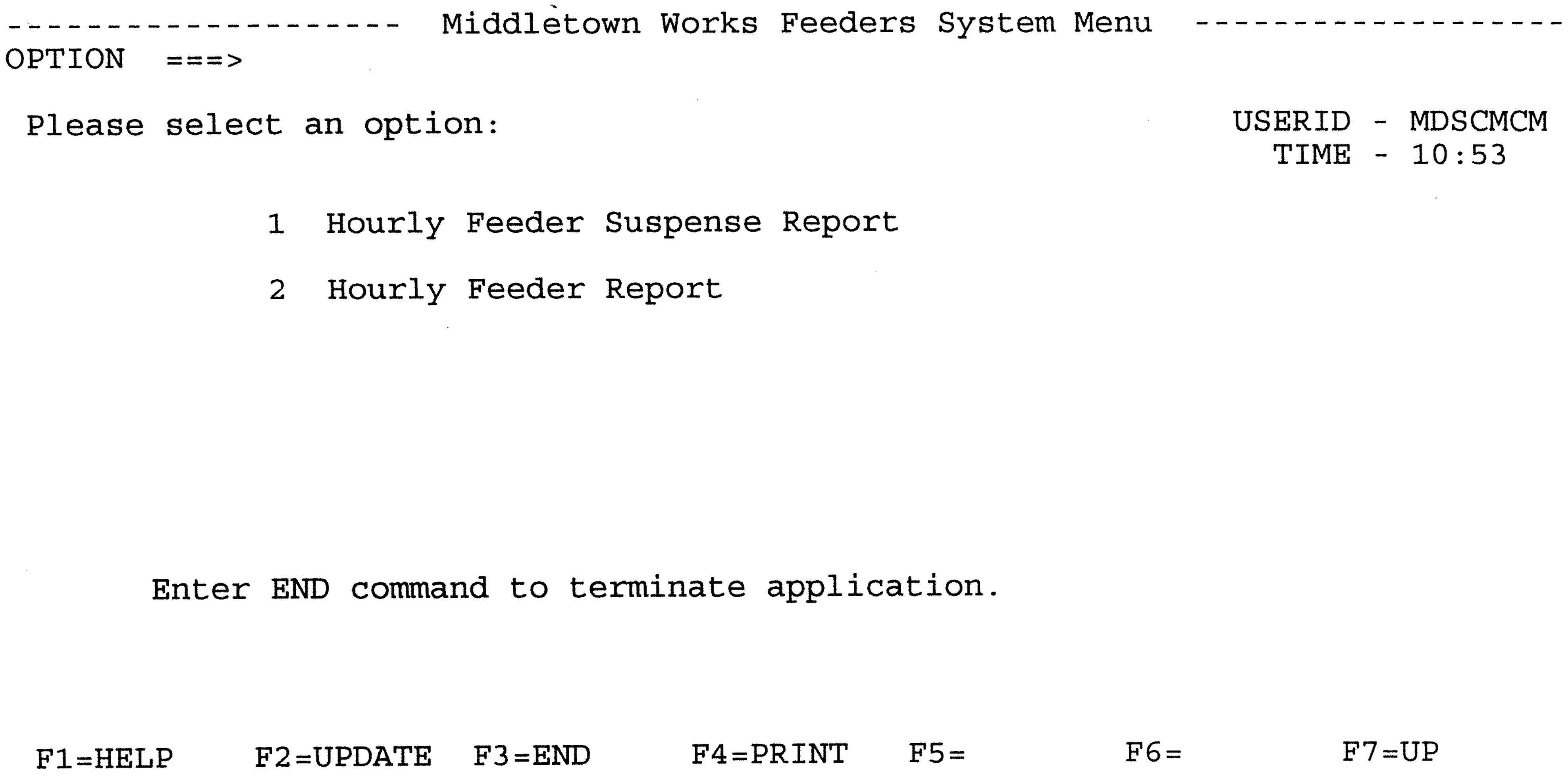

Job Labor
4.1
Retrieve Job Labor Transactions
Process Batch Date and S-Orders
4.2
Manually Entered Job Labor Transactions
4.6
Manually Entered Job Labor Rates
Update Work Order
Work Order
Generate Job Labor Accounts
Job Labor Rates
Shops
General Ledger Accounts
Operating Cost Accounts
Archieve Work Order
4.7
View & Print Job Labor Reports
Verify Valid Accounts
4.4
Re-edit Job Labor Accounts
Work Order History
4.3
Edit Job Labor Suspense Accounts
General Ledger Vouchers
4.5
Load General Ledger Vouchers
Job Labor Suspense Accounts
Job Labor History Accounts
Archieve Job Labor Accounts
Job Labor Accounts

Accounts Payable Process Model

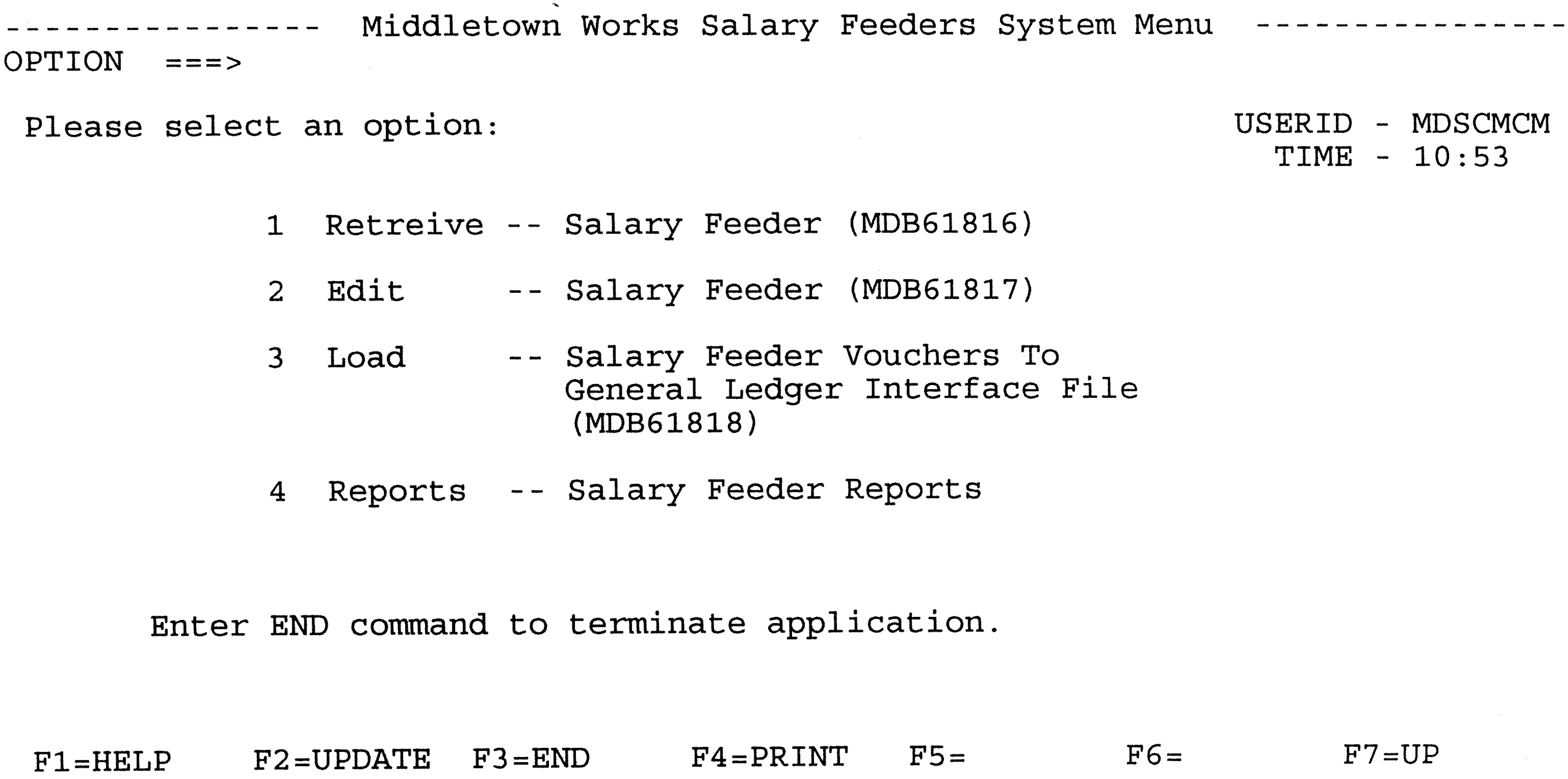

Accounts Payable
5.1
Retrieve Accounts Payable Transactions
Generate Accounts Payable Accounts
Purchase Orders
General Ledger Accounts
Operating Cost Accounts
5.5
View & Print Accounts Payable Reports
General Ledger Vouchers
Verify Valid Accounts
5.3
Re-edit Accounts Payable Accounts
5.4
Load General Ledger Vouchers
5.2
Edit Accounts Payable Suspense Accounts
5.6
Archieve Accounts Payable Accounts
Accounts Payable History
Accounts Payable Suspense Accounts
Accounts Payable Accounts
5.7
Generate 1430 Accounts
1430 Accounts



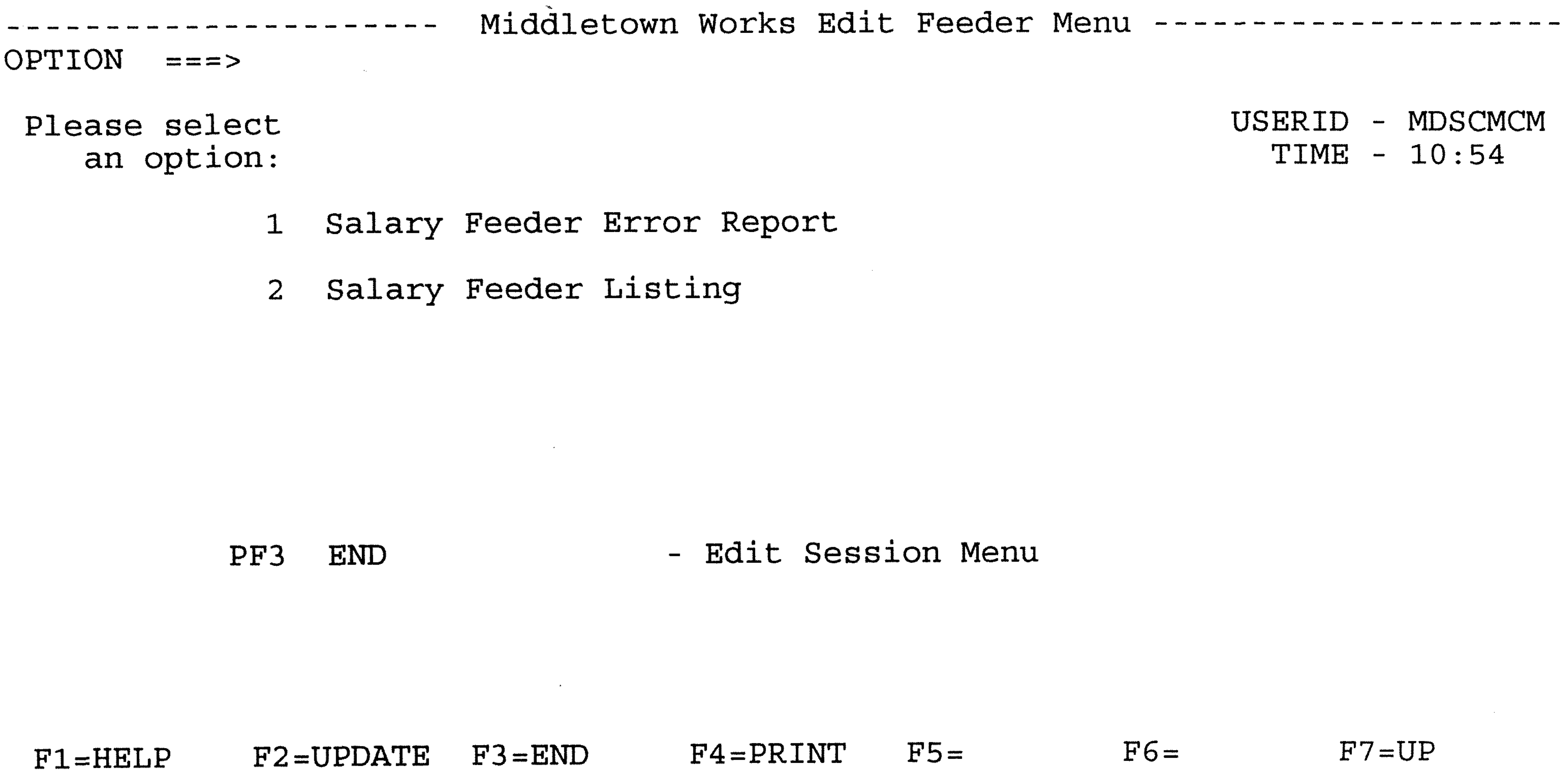
```
----------------  Middletown Works Hourly Feeders System Menu  ----------------
OPTION  ===>

 Please select an option:                                       USERID - MDSCMCM
                                                                  TIME - 10:52

             1  Retreive -- Hourly Feeder (MDB61813)

             2  Edit     -- Hourly Feeder (MDB61814)

             3  Load     -- Hourly Feeder Vouchers To
                            General Ledger Interface File
                            (MDB61815)

             4  Reports  -- Hourly Feeder Reports


       Enter END command to terminate application.


 F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



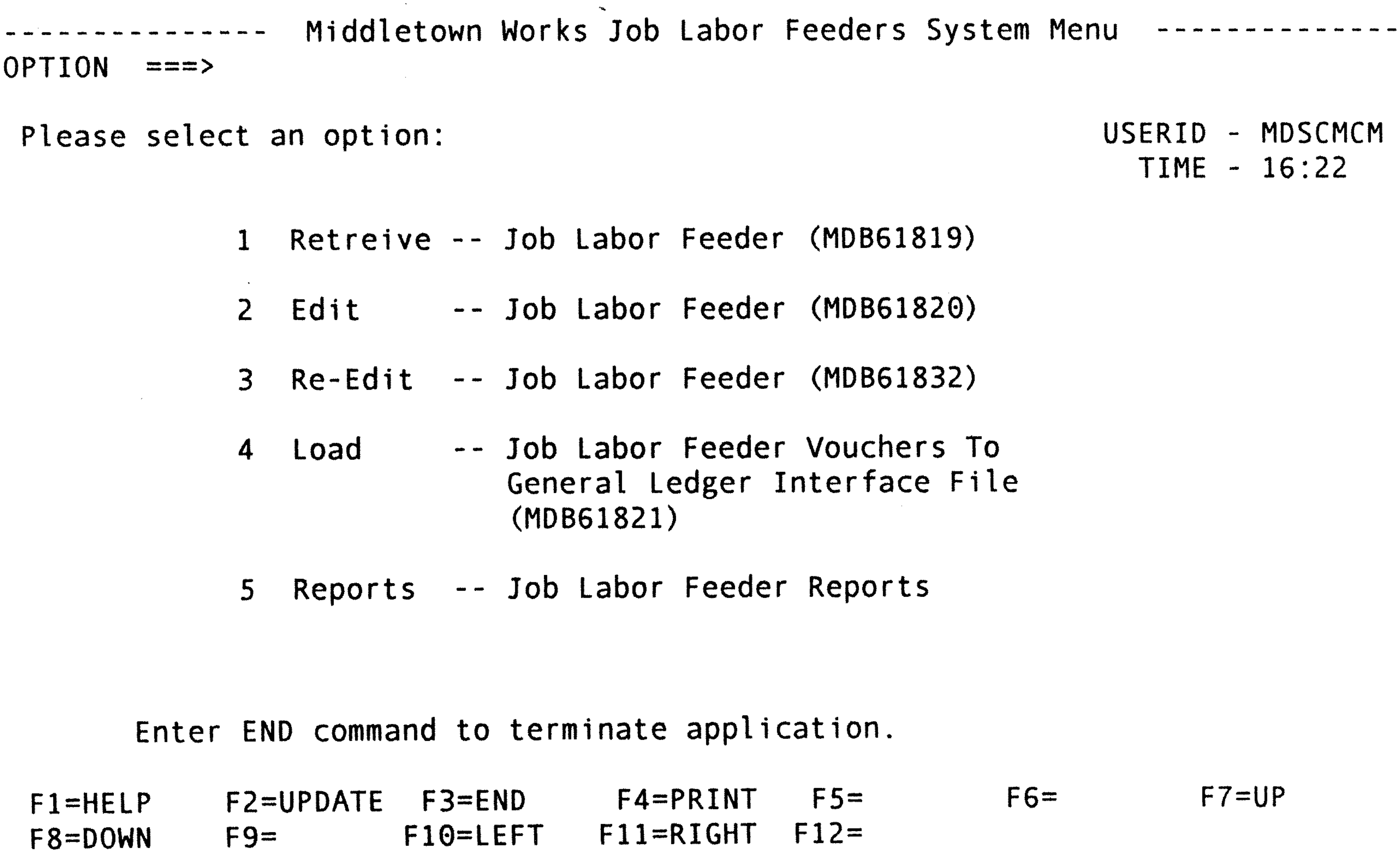
```
--------------------  Middletown Works Feeders System Menu  --------------------
OPTION  ===>

 Please select an option:                                      USERID - MDSCMCM
                                                                 TIME - 10:53

              1  Hourly Feeder Suspense Report

              2  Hourly Feeder Report




       Enter END command to terminate application.



  F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



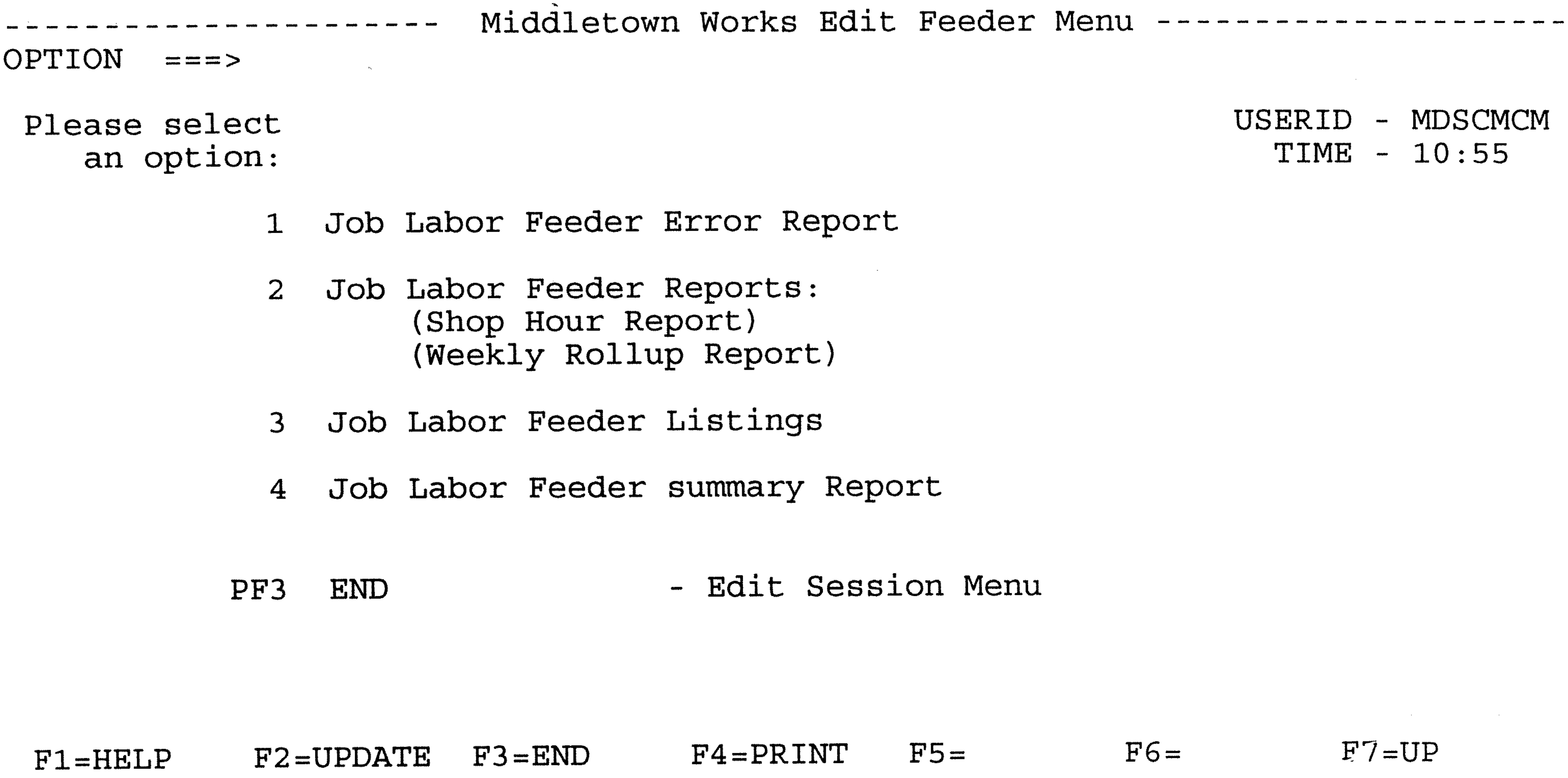
```
----------------  Middletown Works Salary Feeders System Menu  ----------------
OPTION  ===>

 Please select an option:                                      USERID - MDSCMCM
                                                                 TIME - 10:53

              1  Retreive -- Salary Feeder (MDB61816)

              2  Edit     -- Salary Feeder (MDB61817)

              3  Load     -- Salary Feeder Vouchers To
                             General Ledger Interface File
                             (MDB61818)

              4  Reports  -- Salary Feeder Reports


        Enter END command to terminate application.


  F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



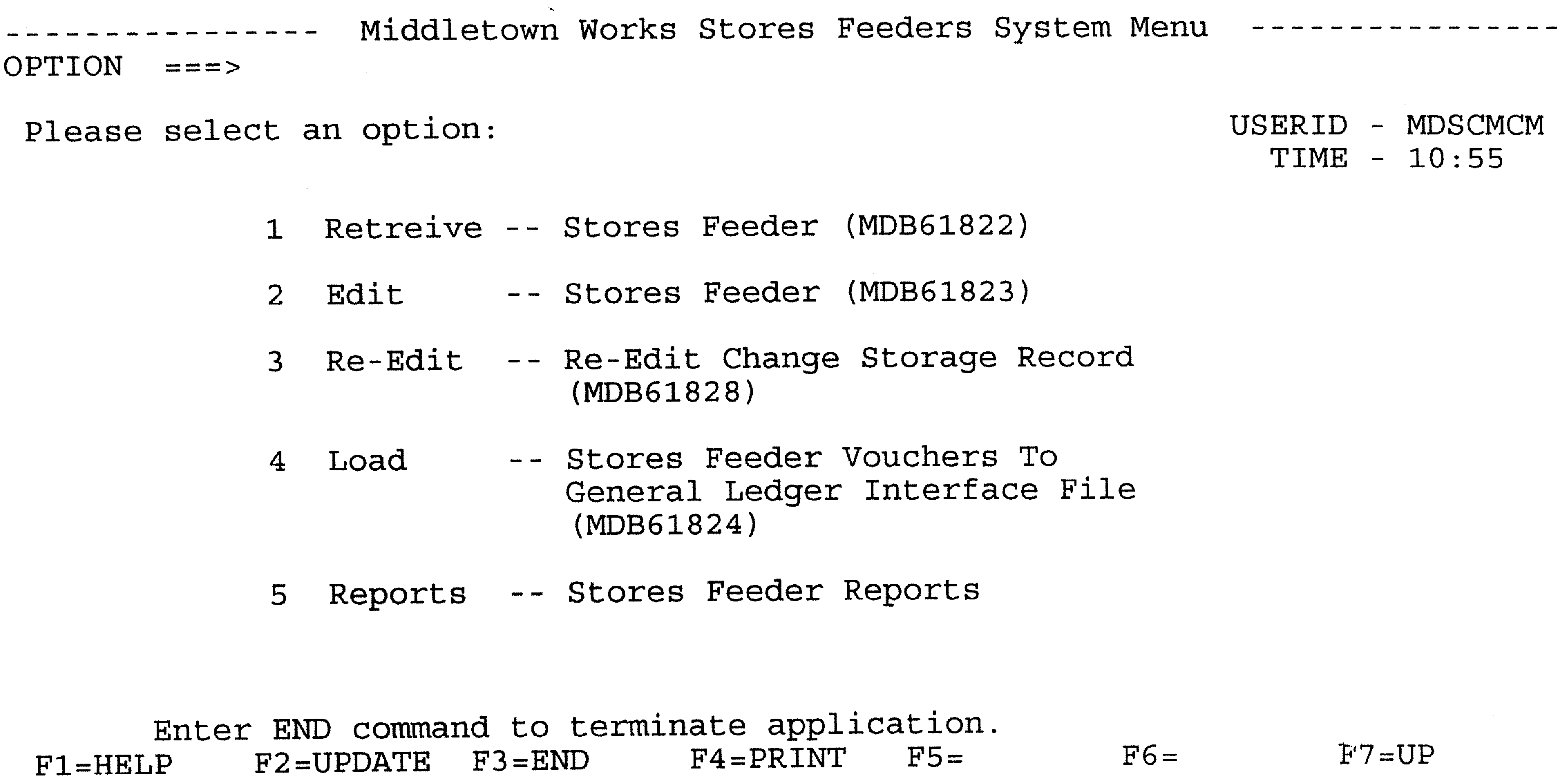
```
---------------------- Middletown Works Edit Feeder Menu ---------------------
OPTION  ===>

 Please select                                                  USERID - MDSCMCM
    an option:                                                    TIME - 10:54

               1  Salary Feeder Error Report

               2  Salary Feeder Listing




           PF3  END              - Edit Session Menu




   F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



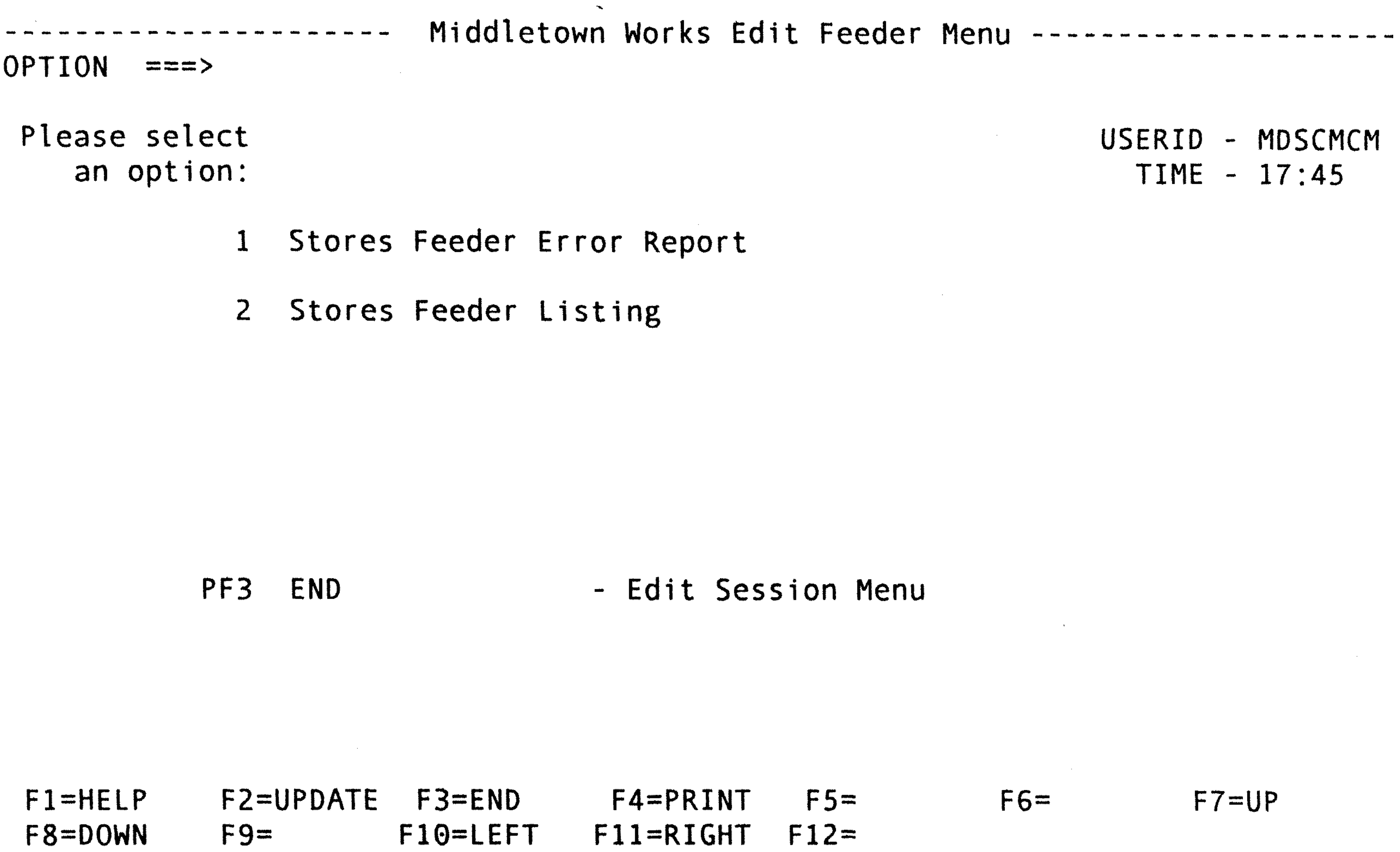
```
---------------  Middletown Works Job Labor Feeders System Menu  -------------
OPTION  ===>

 Please select an option:                                       USERID - MDSCMCM
                                                                  TIME - 16:22

              1  Retreive -- Job Labor Feeder (MDB61819)

              2  Edit     -- Job Labor Feeder (MDB61820)

              3  Re-Edit  -- Job Labor Feeder (MDB61832)

              4  Load     -- Job Labor Feeder Vouchers To
                             General Ledger Interface File
                             (MDB61821)

              5  Reports  -- Job Labor Feeder Reports


        Enter END command to terminate application.

 F1=HELP     F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
 F8=DOWN     F9=       F10=LEFT   F11=RIGHT  F12=
```



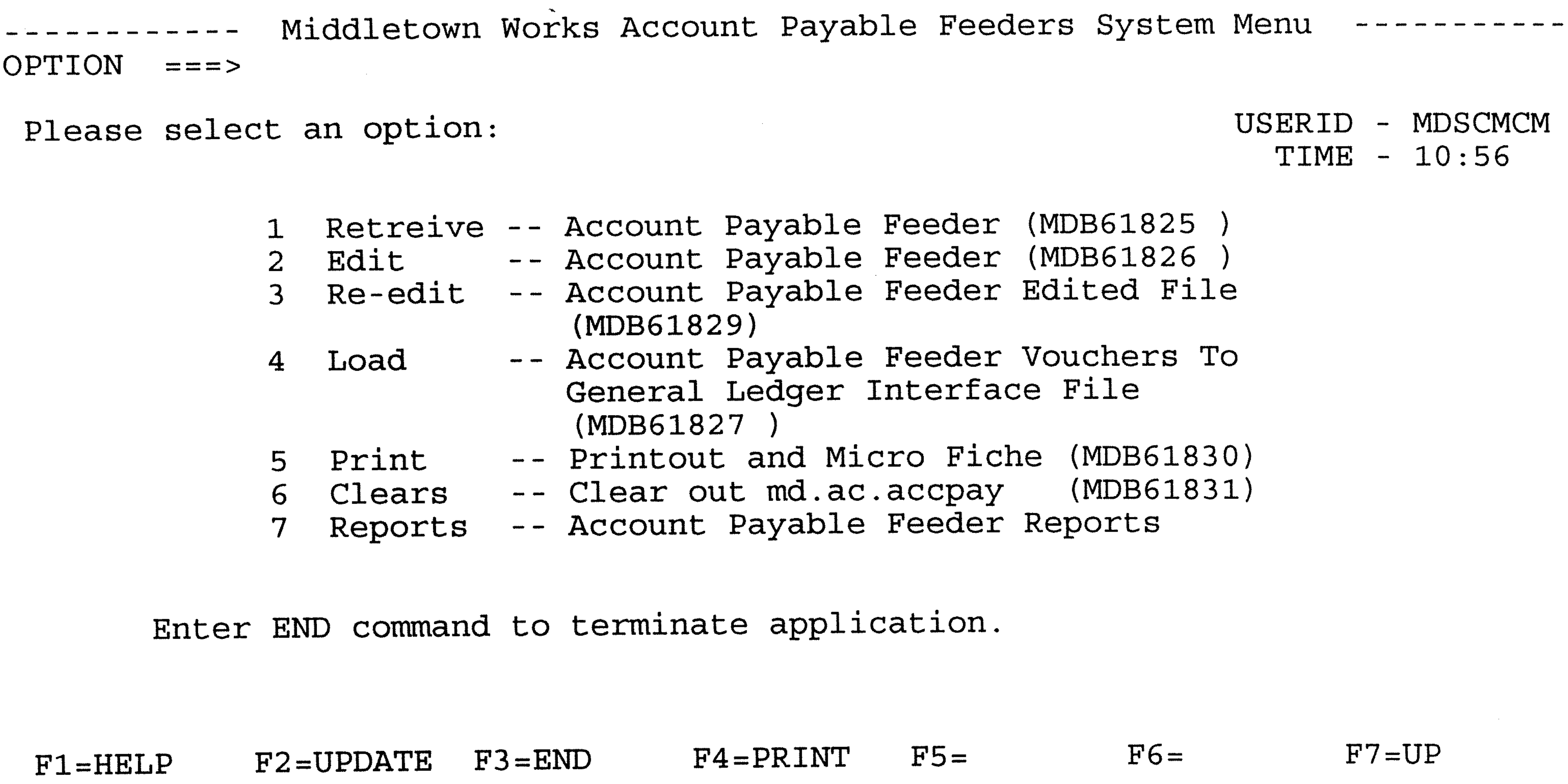
```
---------------------  Middletown Works Edit Feeder Menu  --------------------
OPTION  ===>

 Please select                                                USERID - MDSCMCM
    an option:                                                  TIME - 10:55

              1  Job Labor Feeder Error Report

              2  Job Labor Feeder Reports:
                     (Shop Hour Report)
                     (Weekly Rollup Report)

              3  Job Labor Feeder Listings

              4  Job Labor Feeder summary Report


             PF3  END              - Edit Session Menu



  F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



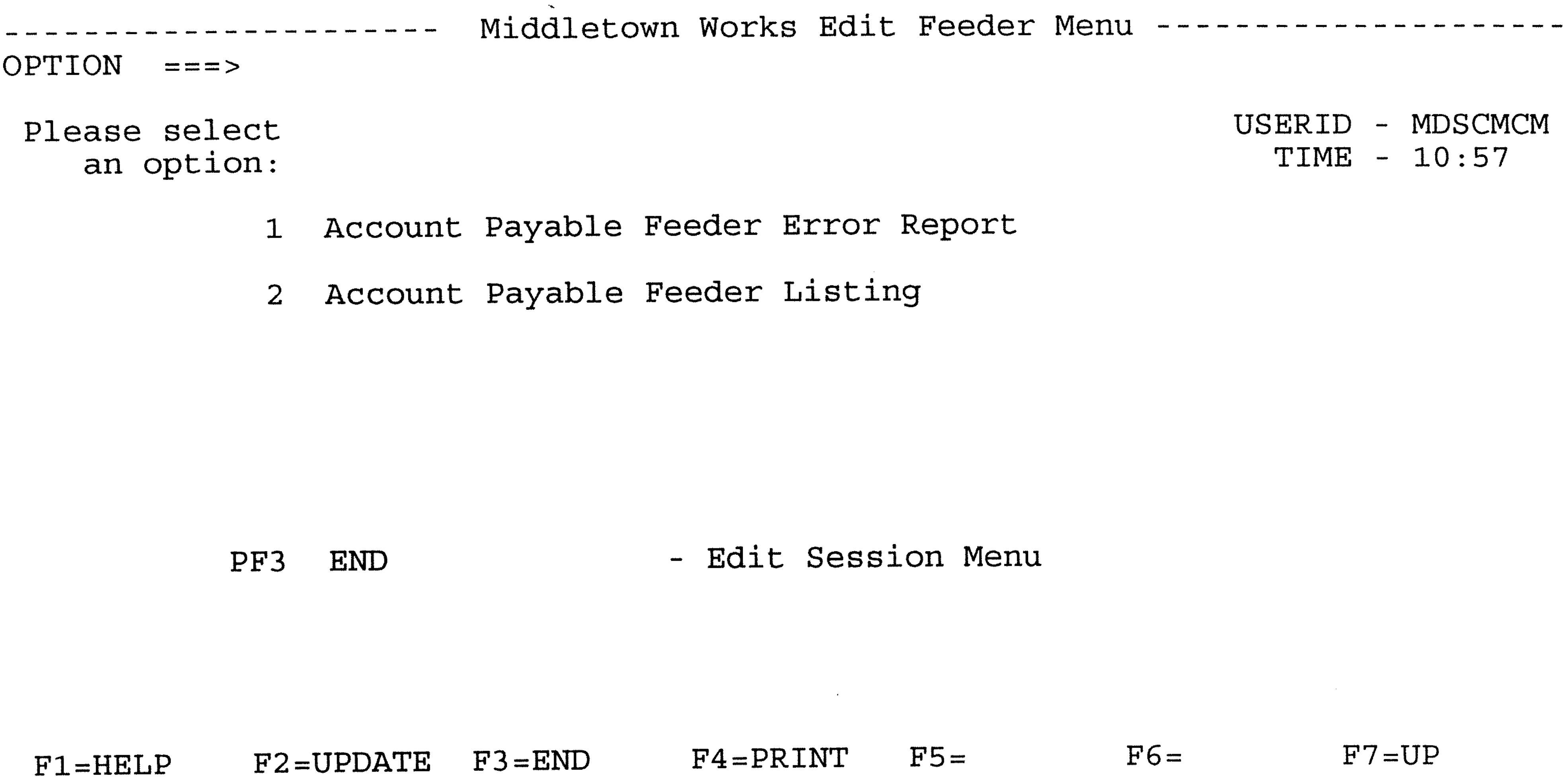
```
---------------  Middletown Works Stores Feeders System Menu  ---------------
OPTION  ===>

 Please select an option:                                      USERID - MDSCMCM
                                                                 TIME - 10:55

              1  Retreive -- Stores Feeder (MDB61822)

              2  Edit     -- Stores Feeder (MDB61823)

              3  Re-Edit  -- Re-Edit Change Storage Record
                             (MDB61828)

              4  Load     -- Stores Feeder Vouchers To
                             General Ledger Interface File
                             (MDB61824)

              5  Reports  -- Stores Feeder Reports


        Enter END command to terminate application.
  F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=         F6=         F7=UP
```



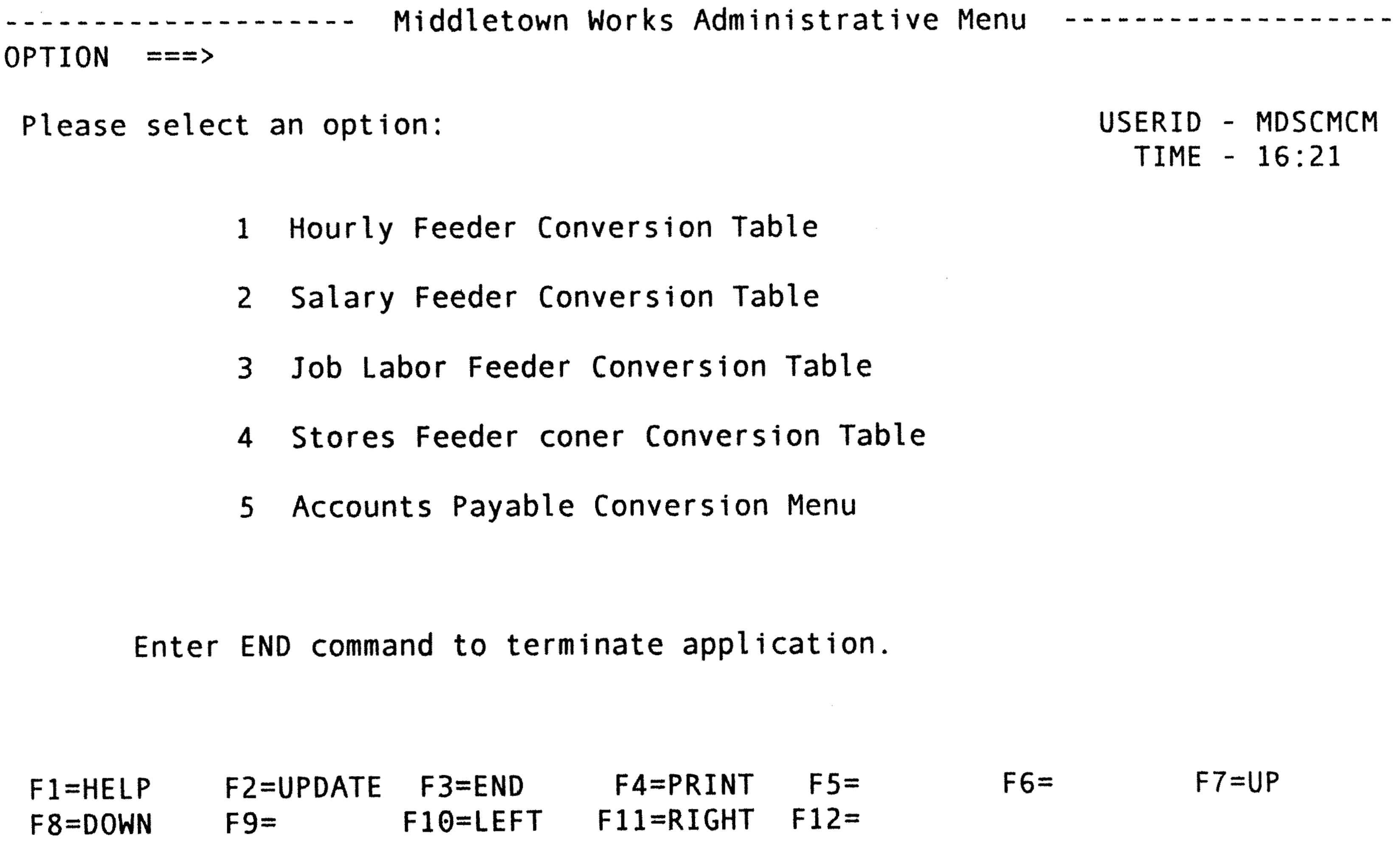
```
----------------------  Middletown Works Edit Feeder Menu ---------------------
OPTION  ===>

 Please select                                                USERID - MDSCMCM
    an option:                                                  TIME - 17:45

              1  Stores Feeder Error Report

              2  Stores Feeder Listing





             PF3  END                - Edit Session Menu





  F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
  F8=DOWN    F9=       F10=LEFT   F11=RIGHT  F12=
```



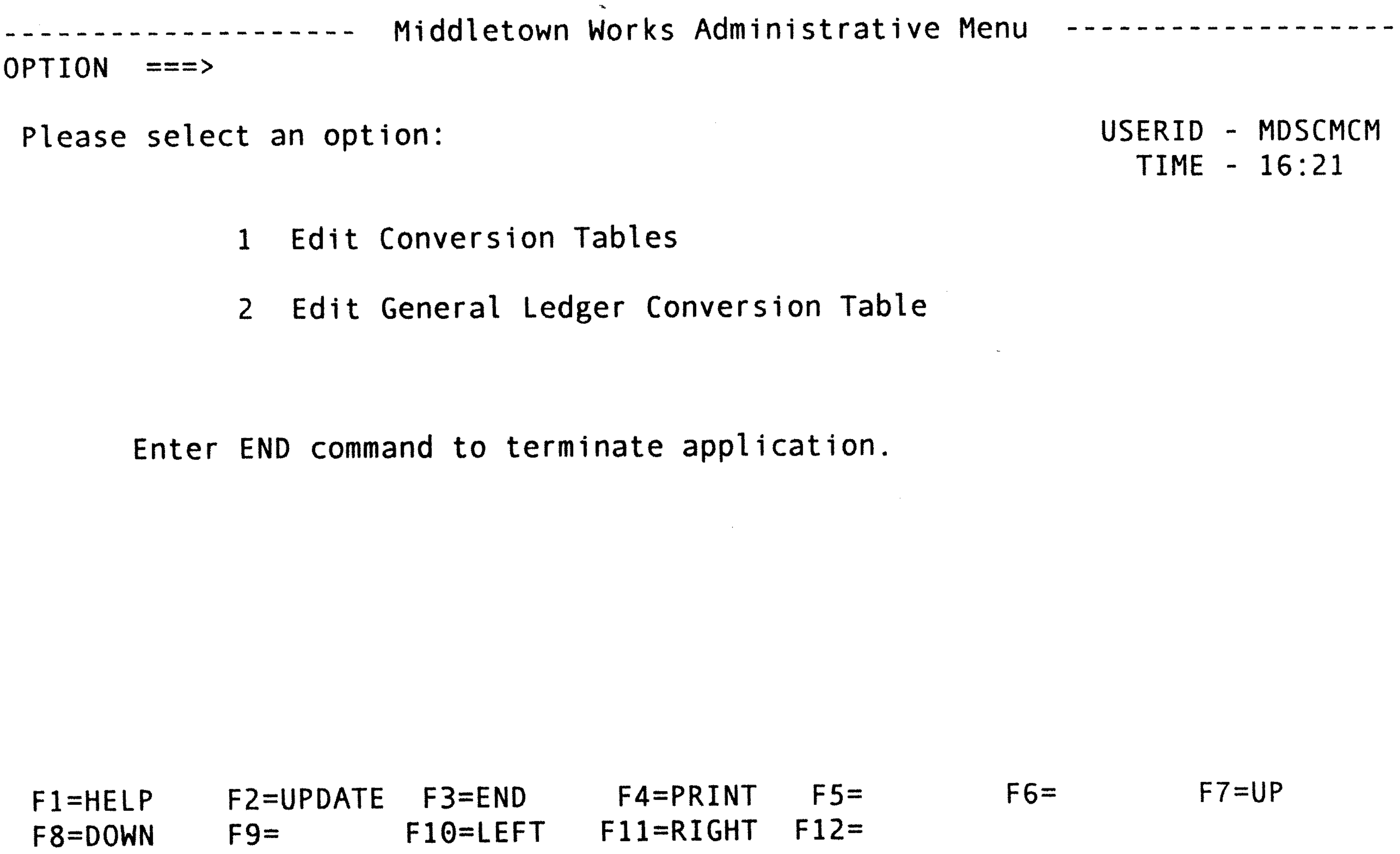
```
------------  Middletown Works Account Payable Feeders System Menu  -----------
OPTION  ===>

 Please select an option:                                      USERID - MDSCMCM
                                                                 TIME - 10:56

             1  Retreive -- Account Payable Feeder (MDB61825 )
             2  Edit     -- Account Payable Feeder (MDB61826 )
             3  Re-edit  -- Account Payable Feeder Edited File
                            (MDB61829)
             4  Load     -- Account Payable Feeder Vouchers To
                            General Ledger Interface File
                            (MDB61827 )
             5  Print    -- Printout and Micro Fiche (MDB61830)
             6  Clears   -- Clear out md.ac.accpay   (MDB61831)
             7  Reports  -- Account Payable Feeder Reports


       Enter END command to terminate application.


  F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



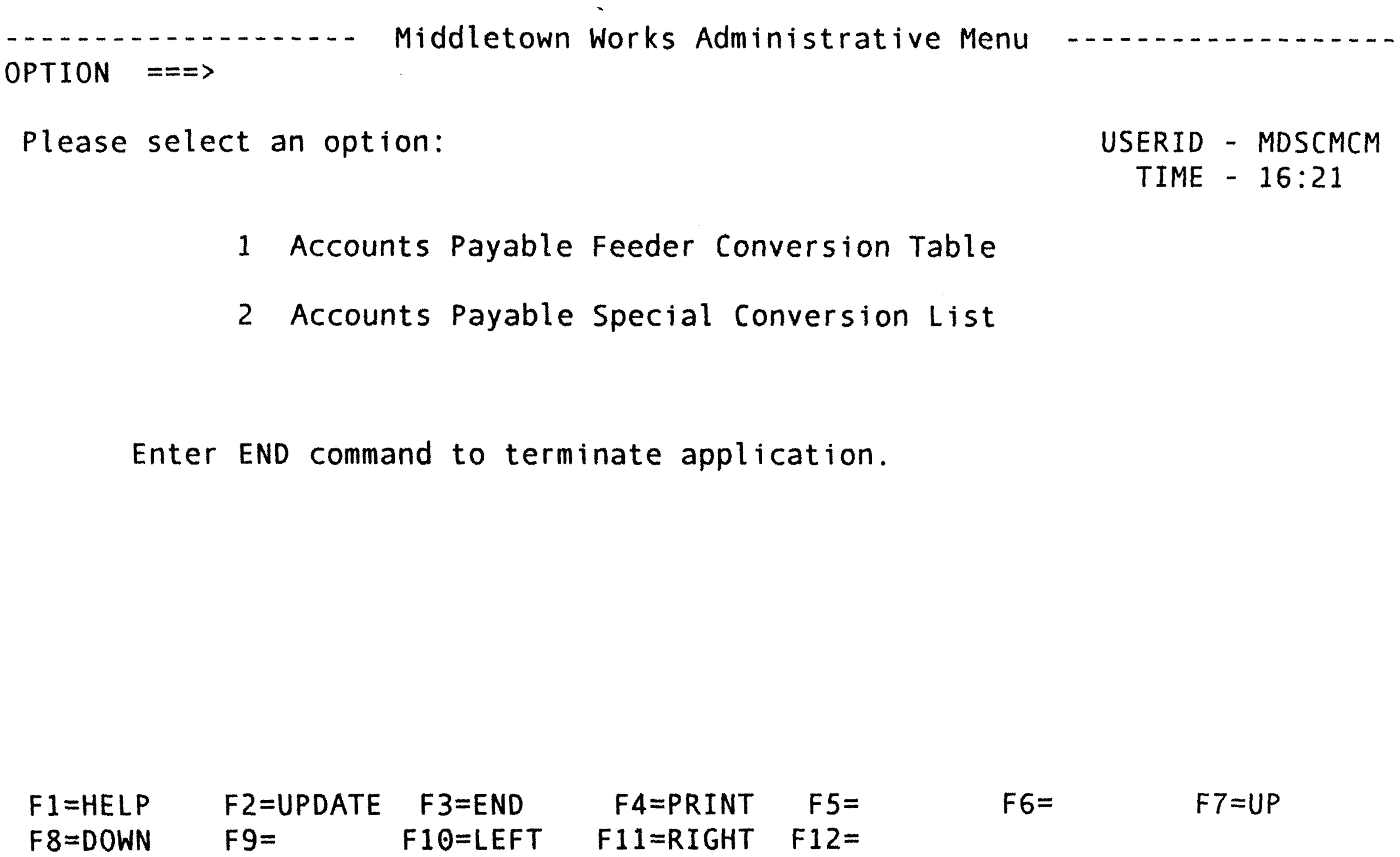
```
---------------------- Middletown Works Edit Feeder Menu ---------------------
OPTION  ===>

 Please select                                                 USERID - MDSCMCM
    an option:                                                   TIME - 10:57

               1  Account Payable Feeder Error Report

               2  Account Payable Feeder Listing




             PF3  END                 - Edit Session Menu




   F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
```



```
-------------------  Middletown Works Administrative Menu  ------------------
OPTION  ===>

 Please select an option:                                   USERID - MDSCMCM
                                                              TIME - 16:21

              1  Hourly Feeder Conversion Table

              2  Salary Feeder Conversion Table

              3  Job Labor Feeder Conversion Table

              4  Stores Feeder coner Conversion Table

              5  Accounts Payable Conversion Menu


        Enter END command to terminate application.


  F1=HELP     F2=UPDATE  F3=END      F4=PRINT   F5=         F6=         F7=UP
  F8=DOWN     F9=         F10=LEFT   F11=RIGHT  F12=
```



```
------------------- Middletown Works Administrative Menu -------------------
OPTION  ===>

 Please select an option:                                   USERID - MDSCMCM
                                                              TIME - 16:21

              1  Edit Conversion Tables

              2  Edit General Ledger Conversion Table


        Enter END command to terminate application.






  F1=HELP     F2=UPDATE  F3=END     F4=PRINT   F5=         F6=         F7=UP
  F8=DOWN     F9=        F10=LEFT   F11=RIGHT  F12=
```



```
-------------------  Middletown Works Administrative Menu  -------------------
OPTION  ===>

 Please select an option:                                       USERID - MDSCMCM
                                                                  TIME - 16:21

              1  Accounts Payable Feeder Conversion Table

              2  Accounts Payable Special Conversion List


         Enter END command to terminate application.








 F1=HELP    F2=UPDATE  F3=END     F4=PRINT   F5=        F6=        F7=UP
 F8=DOWN    F9=        F10=LEFT   F11=RIGHT  F12=
------------------------ THE FEEDER CONTROL DATE CARD ------------------------
----------------------------- Jobname: MDB61813 -----------------------------
OPTION  ===>
                                                                 USERID - MDSCMCM
 Select desired month:                                             TIME - 10:54


            Month       ===> DEC :      JAN FEB MAR APR MAY JUN
                                        JUL AUG SEP OCT NOV DEC







                  PF1 - HELP        PF2 - UPDATE     PF3 - END
  F1=HELP    F2=UPDATE  F3=END      F4=PRINT   F5=        F6=         F7=UP
```

```
                                                     01:01 FRIDAY, NOVEMBER 28, 1996

                         MIDDLETOWN WORKS: FEEDER -- HOURLY PAYROLL
                                        BATCH VOUCHERS

 OBS     FEEDER  GENERAL   COST   EXPENSE    SUB    PROJECT SUB LEDGER  UNITS   DOLLARS     DATE
         SOURCE  ACCOUNT  CENTER    CODE   ACCOUNT              Inv
------------------------------------------------------------------------------------------------

      1    HL       9000             1000             1000              10.00  1000.00  113196
      2    HL       5575    9000     1000    0000                       10.00  1000.00  113196
      3    HL       5575    9000     2000    0000                       10.00  1000.00  113196
      4    HL       5575    9000     3000    0000                       10.00  1000.00  113196
------------------------------------------------------------------------------------------------
                                                                       40.00   4000.00
```

FIGURE 1

```
                                          01:01 FRIDAY,  NOVEMBER 28, 1996

                               FEEDER -- HOURLY PAYROLL
                        VOUCHERS WHICH WILL NOT POST TO THE GENERAL LEDGER
                        BECAUSE THE RECORDS ARE NOT FOUND IN THE MASTER KEY

OBS  GENERAL   COST   EXPENSE   SUB    PROJECT   PROD    UNIT   DOLLARS  SOURCE  SOURCE   SOURCE    ERROR
     ACCOUNT  CENTER   CODE   ACCOUNT            Inv                     CENTER  EXPENSE   SUB      CODE
------------------------------------------------------------------------------------------ -------- ---------

  1    9000            1000             1000             10.00  1000.00    9000    1000      0000 NOT FOUND
  2    5575    9000    1000    0000                      10.00  1000.00    9000    1000      0000 NOT FOUND
  3    5575    9000    2000    0000                      10.00  1000.00    9000    2000      0000 NOT FOUND
  4    5575    9000    3000    0000                      10.00  1000.00    9000    3000      0000 NOT FOUND
```

FIGURE 2

```
                                                   01:01 FRIDAY,  NOVEMBER 28, 1996

                             HOURLY PAYROLL EXPENSE ITEM NUMBER NOT FOUND
                             IN COST SHEET DATABASE -- EXPENSE ITEM TABLE

  OBS   GENERAL   COST   EXPENSE    SUB    PROJECT   PROD    UNIT   DOLLARS   SOURCE  SOURCE   SOURCE
        ACCOUNT  CENTER   CODE    ACCOUNT            Inv                      CENTER  EXPENSE   SUB
------------------------------------------------------------------------------------------ --------

     1    9000             1000              1000            10.00  1000.00     9000    1000    0000
     2    5575     9000    1000     0000                     10.00  1000.00     9000    1000    0000
     3    5575     9000    2000     0000                     10.00  1000.00     9000    2000    0000
     4    5575     9000    3000     0000                     10.00  1000.00     9000    3000    0000
```

FIGURE 3

```
                                                  01:01 FRIDAY, NOVEMBER 28, 1996

                         MIDDLETOWN WORKS: FEEDER -- SALARY PAYROLL
                                      BATCH VOUCHERS

  OBS   FEEDER  GENERAL   COST  EXPENSE    SUB    PROJECT SUB LEDGER  UNITS    DOLLARS    DATE
        SOURCE  ACCOUNT  CENTER   CODE   ACCOUNT              Inv
-----------------------------------------------------------------------------------------------

     1    SL       9000            1000             1000              10.00  1000.00  113196
     2    SL       5575    9000    1000    0000                       10.00  1000.00  113196
     3    SL       5575    9000    2000    0000                       10.00  1000.00  113196
     4    SL       5575    9000    3000    0000                       10.00  1000.00  113196
-----------------------------------------------------------------------------------------------
                                                                     40.00   4000.00
```

FIGURE 4

```
                                                                 01:01 FRIDAY,  NOVEMBER 28, 1996

                                           FEEDER -- SALARY PAYROLL
                              VOUCHERS WHICH WILL NOT POST TO THE GENERAL LEDGER
                             BECAUSE THE RECORDS ARE NOT FOUND IN THE MASTER KEY

 OBS   GENERAL   COST    EXPENSE    SUB   PROJECT    PROD     UNIT   DOLLARS   SOURCE  SOURCE   SOURCE    ERROR
       ACCOUNT  CENTER    CODE   ACCOUNT             Inv                      CENTER   EXPENSE   SUB       CODE
----------------------------------------------------------------------------------------------- -------- ---------

    1     9000              1000              1000             10.00  1000.00     9000    1000     0000 NOT FOUND
    2     5575     9000     1000     0000                      10.00  1000.00     9000    1000     0000 NOT FOUND
    3     5575     9000     2000     0000                      10.00  1000.00     9000    2000     0000 NOT FOUND
    4     5575     9000     3000     0000                      10.00  1000.00     9000    3000     0000 NOT FOUND
```

FIGURE 5

```
                                                                     01:01 FRIDAY,  NOVEMBER 28, 1996

                                    SALARY PAYROLL EXPENSE ITEM NUMBER NOT FOUND
                                    IN COST SHEET DATABASE -- EXPENSE ITEM TABLE

   OBS    GENERAL   COST    EXPENSE    SUB    PROJECT   PROD     UNIT   DOLLARS  SOURCE   SOURCE   SOURCE
          ACCOUNT  CENTER    CODE    ACCOUNT             Inv                     CENTER   EXPENSE    SUB
-------------------------------------------------------------------------------------------------- --------

        1     9000              1000             1000             10.00  1000.00    9000     1000     0000
        2     5575     9000     1000    0000                      10.00  1000.00    9000     1000     0000
        3     5575     9000     2000    0000                      10.00  1000.00    9000     2000     0000
        4     5575     9000     3000    0000                      10.00  1000.00    9000     3000     0000
```

FIGURE 6

```
                                                                   01:01 FRIDAY, NOVEMBER 28, 1996

                                   MIDDLETOWN WORKS: FEEDER -- JOB LABOR
                                              BATCH VOUCHERS

  OBS     FEEDER  GENERAL   COST   EXPENSE    SUB    PROJECT  SUB LEDGER  UNIT    DOLLARS    DATE
          SOURCE  ACCOUNT  CENTER    CODE   ACCOUNT              Inv
----------------------------------------------------------------------------------------------------

     1     JL       9000              1000             1000              10.00  1000.00  113196
     2     JL       5575     9000     1000    0000                       10.00  1000.00  113196
     3     JL       5575     9000     2000    0000                       10.00  1000.00  113196
     4     JL       5575     9000     3000    0000                       10.00  1000.00  113196
----------------------------------------------------------------------------------------------------
                                                                        40.00   4000.00
```

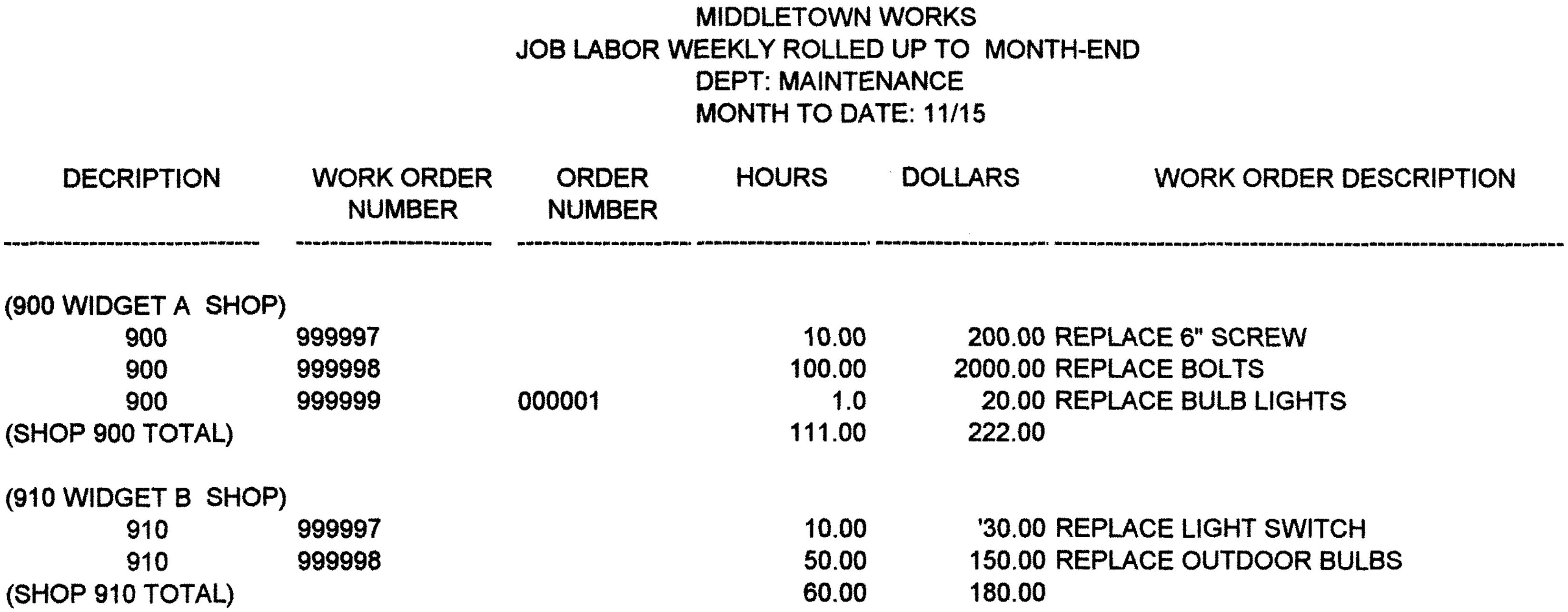
FIGURE 7

```
                                                        01:01 FRIDAY,  NOVEMBER 28, 1996

                                          FEEDER -- JOB LABOR
                               VOUCHERS WHICH WILL NOT POST TO THE GENERAL LEDGER
                               BECAUSE THE RECORDS ARE NOT FOUND IN THE MASTER KEY

OBS  GENERAL  COST     EXPENSE   SUB      PROJECT    PROD      UNIT     DOLLARS   SOURCE   SOURCE    SOURCE     ERROR
     ACCOUNT  CENTER   CODE     ACCOUNT              Inv                          CENTER   EXPENSE   SUB        CODE
-------------------------------------------------------------------------------------------------------- -------- ---------

  1    9000             1000               1000              10.00   1000.00     9000     1000       0000 NOT FOUND
  2    5575    9000     1000      0000                       10.00   1000.00     9000     1000       0000 NOT FOUND
  3    5575    9000     2000      0000                       10.00   1000.00     9000     2000       0000 NOT FOUND
  4    5575    9000     3000      0000                       10.00   1000.00     9000     3000       0000 NOT FOUND
```

FIGURE 8

```
                                                                     01:01 FRIDAY,  NOVEMBER 28, 1996

                                      JOB LABOR EXPENSE ITEM NUMBER NOT FOUND
                                   IN COST SHEET DATABASE -- EXPENSE ITEM TABLE

   OBS      GENERAL   COST    EXPENSE    SUB    PROJECT   PROD     UNIT   DOLLARS   SOURCE   SOURCE   SOURCE
          ACCOUNT  CENTER    CODE    ACCOUNT            Inv                       CENTER  EXPENSE    SUB
-------------------------------------------------------------------------------------------------- --------

       1      9000             1000              1000             10.00  1000.00     9000    1000     0000
       2      5575    9000     1000     0000                      10.00  1000.00     9000    1000     0000
       3      5575    9000     2000     0000                      10.00  1000.00     9000    2000     0000
       4      5575    9000     3000     0000                      10.00  1000.00     9000    3000     0000
```

FIGURE 9



MIDDLETOWN WORKS
JOB LABOR WEEKLY ROLLED UP TO MONTH-END
DEPT: MAINTENANCE
MONTH TO DATE: 11/15

DECRIPTION	WORK ORDER NUMBER	ORDER NUMBER	HOURS	DOLLARS	WORK ORDER DESCRIPTION
(900 WIDGET A SHOP)					
900	999997		10.00	200.00	REPLACE 6" SCREW
900	999998		100.00	2000.00	REPLACE BOLTS
900	999999	000001	1.0	20.00	REPLACE BULB LIGHTS
(SHOP 900 TOTAL)			111.00	222.00	
(910 WIDGET B SHOP)					
910	999997		10.00	'30.00	REPLACE LIGHT SWITCH
910	999998		50.00	150.00	REPLACE OUTDOOR BULBS
(SHOP 910 TOTAL)			60.00	180.00	

FIGURE 11

```
                                                   01:01 FRIDAY, NOVEMBER 28, 1996

                     MIDDLETOWN WORKS: FEEDER -- STORES
                              BATCH VOUCHERS

 OBS    FEEDER  GENERAL   COST   EXPENSE   SUB    PROJECT  SUB LEDGER  UNITS    DOLLARS    DATE
        SOURCE  ACCOUNT  CENTER   CODE   ACCOUNT               Inv
----------------------------------------------------------------------------------------------

    1     ST      9000             1000            1000                 10.00  1000.00  113196
    2     ST      5575    9000     1000    0000                         10.00  1000.00  113196
    3     ST      5575    9000     2000    0000                         10.00  1000.00  113196
    4     ST      5575    9000     3000    0000                         10.00  1000.00  113196
----------------------------------------------------------------------------------------------
                                                                       40.00   4000.00
```

FIGURE 13

01:01 FRIDAY, NOVEMBER 28, 1996

DEPARTMENT COSTS BEFORE DISTRIBUTION
PERIODS ENDS: 11/31/96'

DEPT.	DEPT. NAME	RATES	A/C HOURS	A/C DOLLAR	SELF HOURS	OTHER HOURS	OTHER DOLLARS	TOTAL HOURS	DIST. HOUR	DIST. DOLLARS
900	WIDGET A	1.00	00.00	00.00	10.00	1000.00	1000.00	1010.00	1000.00	1000.00
910	WIDGET B	2.00	00.00	00.00	11.00	500.00	1000.00	511.00	500.00	500.00
920	WIDGET C	3.00	00.00	00.00	100.00	1000.00	3000.00	1100.00	1000.00	1000.00
930	WIDGET D	4.00	00.00	00.00	0.00	200.00	800.00	200.00	200.00	200.00
940	WIDGET E	5.00	00.00	00.00	0.00	400.00	2000.00	400.00	400.00	400.00
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------
			00.00	00.00	121.00	3100.00	7800.00	3221.00	3100.00	7800.00

FIGURE 12

```
                                                                          01:01 FRIDAY,  NOVEMBER 28, 1996

                                                   FEEDER -- STORES
                                       VOUCHERS WHICH WILL NOT POST TO THE GENERAL LEDGER
                                      BECAUSE THE RECORDS ARE NOT FOUND IN THE MASTER KEY

 OBS   GENERAL   COST    EXPENSE    SUB    PROJECT   PROD     UNIT    DOLLARS  SOURCE   SOURCE   SOURCE    ERROR
       ACCOUNT  CENTER    CODE    ACCOUNT             Inv                      CENTER   EXPENSE    SUB     CODE
------------------------------------------------------------------------------------------------ -------- ---------

    1      9000              1000              1000             10.00  1000.00    9000     1000     0000 NOT FOUND
    2      5575     9000     1000     0000                      10.00  1000.00    9000     1000     0000 NOT FOUND
    3      5575     9000     2000     0000                      10.00  1000.00    9000     2000     0000 NOT FOUND
    4      5575     9000     3000     0000                      10.00  1000.00    9000     3000     0000 NOT FOUND
```

FIGURE 14

```
                                          01:01 FRIDAY, NOVEMBER 28, 1996

                    MIDDLETOWN WORKS: FEEDER -- ACCTS PAYABLE
                                 BATCH VOUCHERS

OBS    FEEDER  GENERAL   COST   EXPENSE    SUB    PROJECT  SUB LEDGER  UNITS   DOLLARS   DATE
       SOURCE  ACCOUNT  CENTER   CODE    ACCOUNT              Inv
---------------------------------------------------------------------------------------------

   1     AP      9000            1000             1000                10.00   1000.00  113196
   2     AP      5575    9000    1000     0000                        10.00   1000.00  113196
   3     AP      5575    9000    2000     0000                        10.00   1000.00  113196
   4     AP      5575    9000    3000     0000                        10.00   1000.00  113196
---------------------------------------------------------------------------------------------
                                                                     40.00   4000.00
```

FIGURE 16

```
                                                                  01:01 FRIDAY,  NOVEMBER 28, 1996

                                            FEEDER -- ACCT PAYABLE
                              VOUCHERS WHICH WILL NOT POST TO THE GENERAL LEDGER
                             BECAUSE THE RECORDS ARE NOT FOUND IN THE MASTER KEY

 OBS  GENERAL    COST    EXPENSE    SUB    PROJECT   PROD     UNIT    DOLLARS   SOURCE   SOURCE   SOURCE    ERROR
      ACCOUNT   CENTER    CODE    ACCOUNT             Inv                       CENTER  EXPENSE    SUB      CODE
------------------------------------------------------------------------------------------------- -------- ---------

    1     9000              1000              1000              10.00 1000.00      9000     1000     0000 NOT FOUND
    2     5575     9000     1000     0000                       10.00 1000.00      9000     1000     0000 NOT FOUND
    3     5575     9000     2000     0000                       10.00 1000.00      9000     2000     0000 NOT FOUND
    4     5575     9000     3000     0000                       10.00 1000.00      9000     3000     0000 NOT FOUND
```

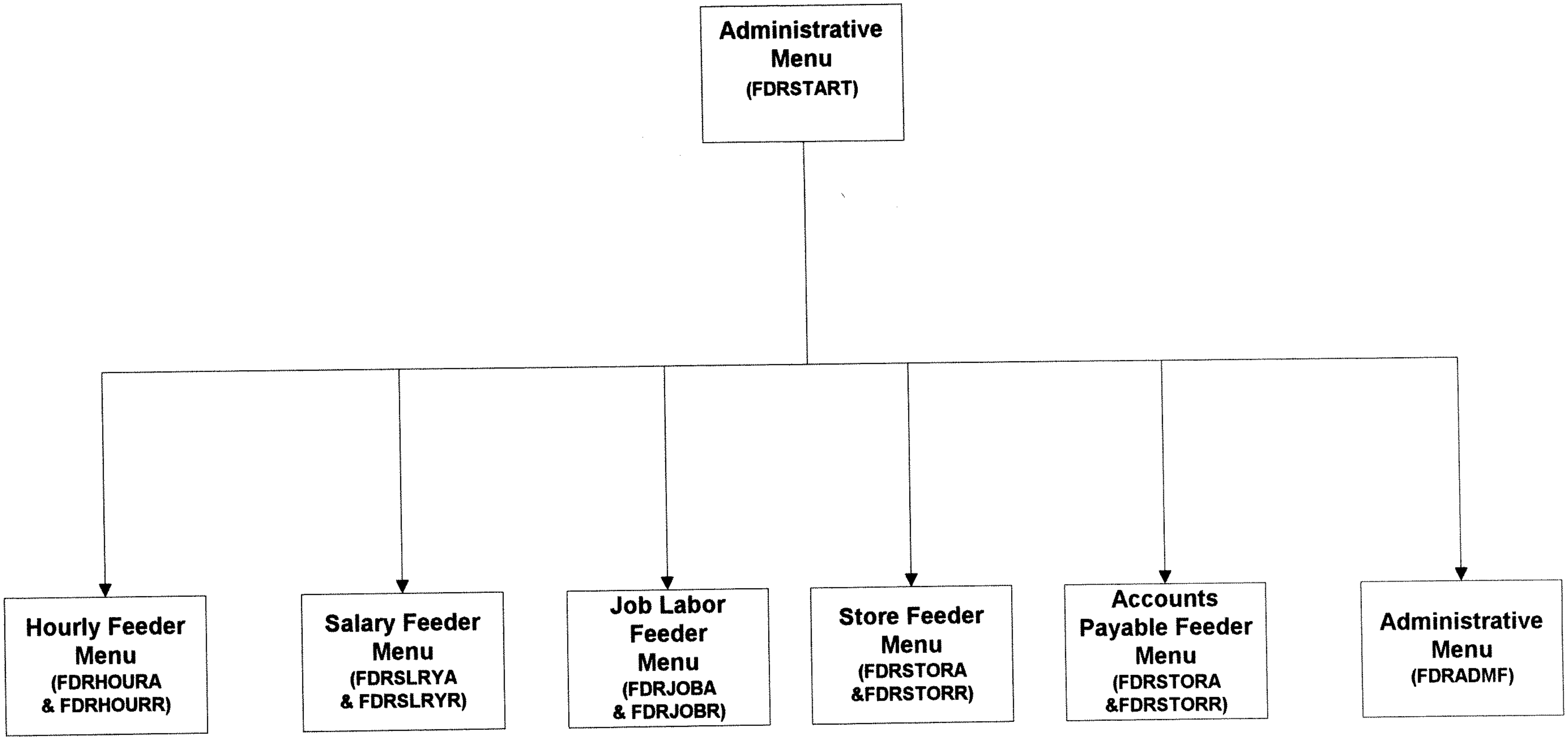
FIGURE 17

```
                                                      01:01 FRIDAY,  NOVEMBER 28, 1996

                         ACCTS PAYABLE EXPENSE ITEM NUMBER NOT FOUND
                         IN COST SHEET DATABASE -- EXPENSE ITEM TABLE

OBS    GENERAL  COST    EXPENSE   SUB    PROJECT  PROD    UNIT   DOLLARS   SOURCE  SOURCE   SOURCE
       ACCOUNT  CENTER  CODE    ACCOUNT           Inv                      CENTER  EXPENSE  SUB
-------------------------------------------------------------------------------------- --------

    1    9000             1000             1000            10.00  1000.00     9000    1000     0000
    2    5575    9000     1000    0000                     10.00  1000.00     9000    1000     0000
    3    5575    9000     2000    0000                     10.00  1000.00     9000    2000     0000
    4    5575    9000     3000    0000                     10.00  1000.00     9000    3000     0000
```

FIGURE 18

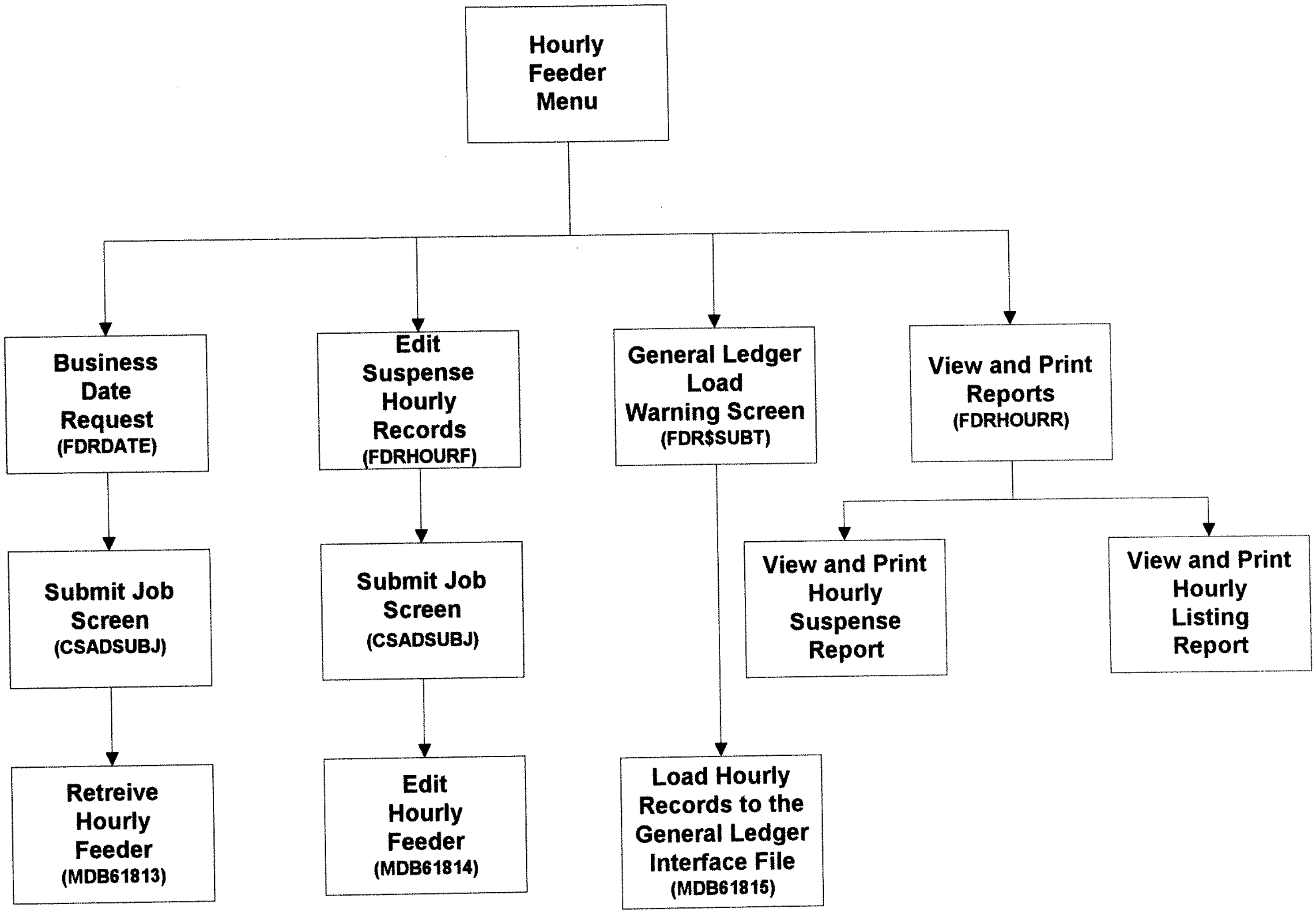
Hierarchy Chart

Middletown Works Feeders System


Administrative Menu (FDRSTART)
Hourly Feeder Menu (FDRHOURA & FDRHOURR)
Salary Feeder Menu (FDRSLRYA & FDRSLRYR)
Job Labor Feeder Menu (FDRJOBA & FDRJOBR)
Store Feeder Menu (FDRSTORA &FDRSTORR)
Accounts Payable Feeder Menu (FDRSTORA &FDRSTORR)
Administrative Menu (FDRADMF)

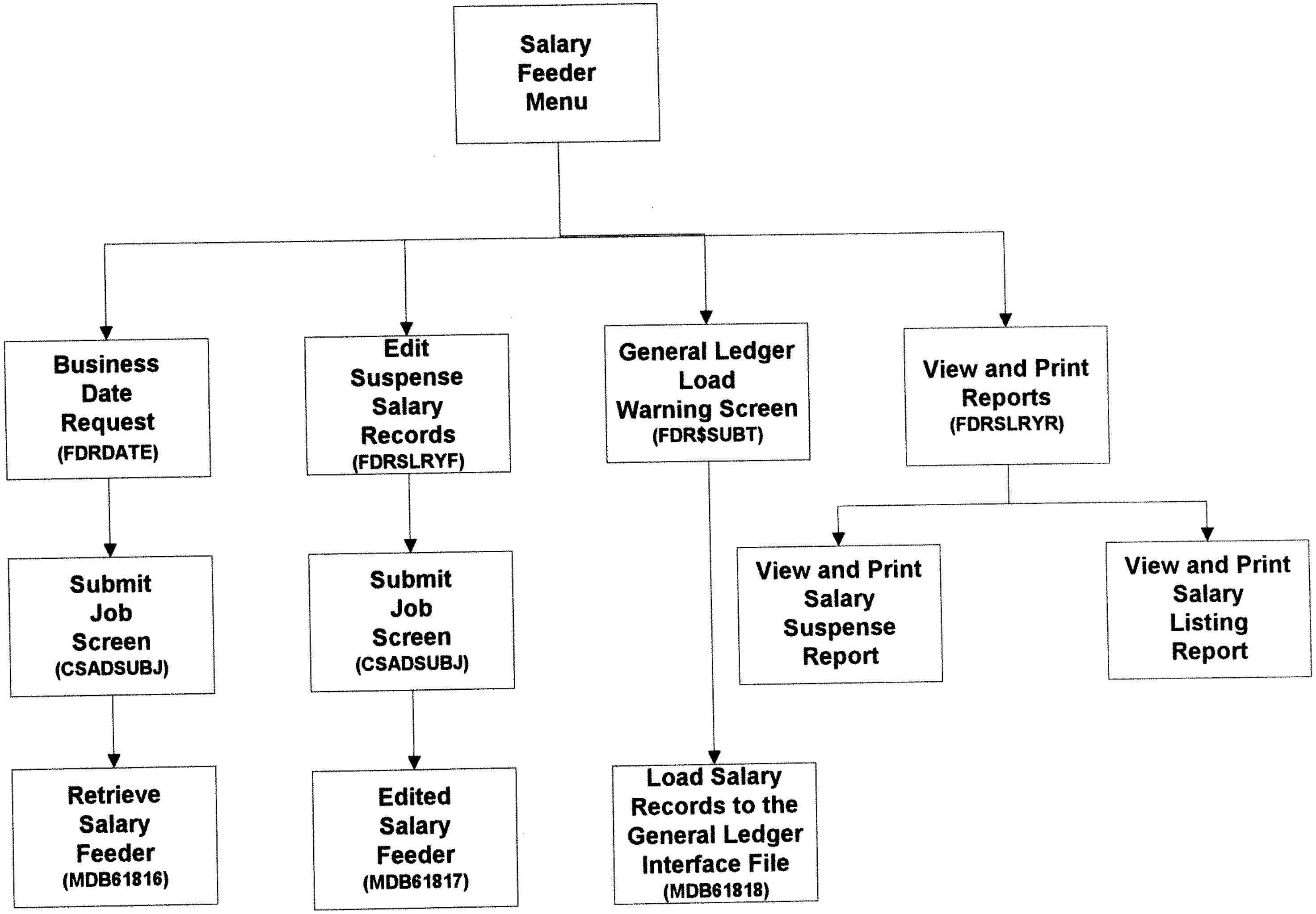
Hierarchy Chart

Middletown Works Feeders System


Hourly Feeder Menu
Business Date Request (FDRDATE)
Edit Suspense Hourly Records (FDRHOURF)
General Ledger Load Warning Screen (FDR$SUBT)
View and Print Reports (FDRHOURR)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
View and Print Hourly Suspense Report
View and Print Hourly Listing Report
Retreive Hourly Feeder (MDB61813)
Edit Hourly Feeder (MDB61814)
Load Hourly Records to the General Ledger Interface File (MDB61815)

Hierarchy Chart

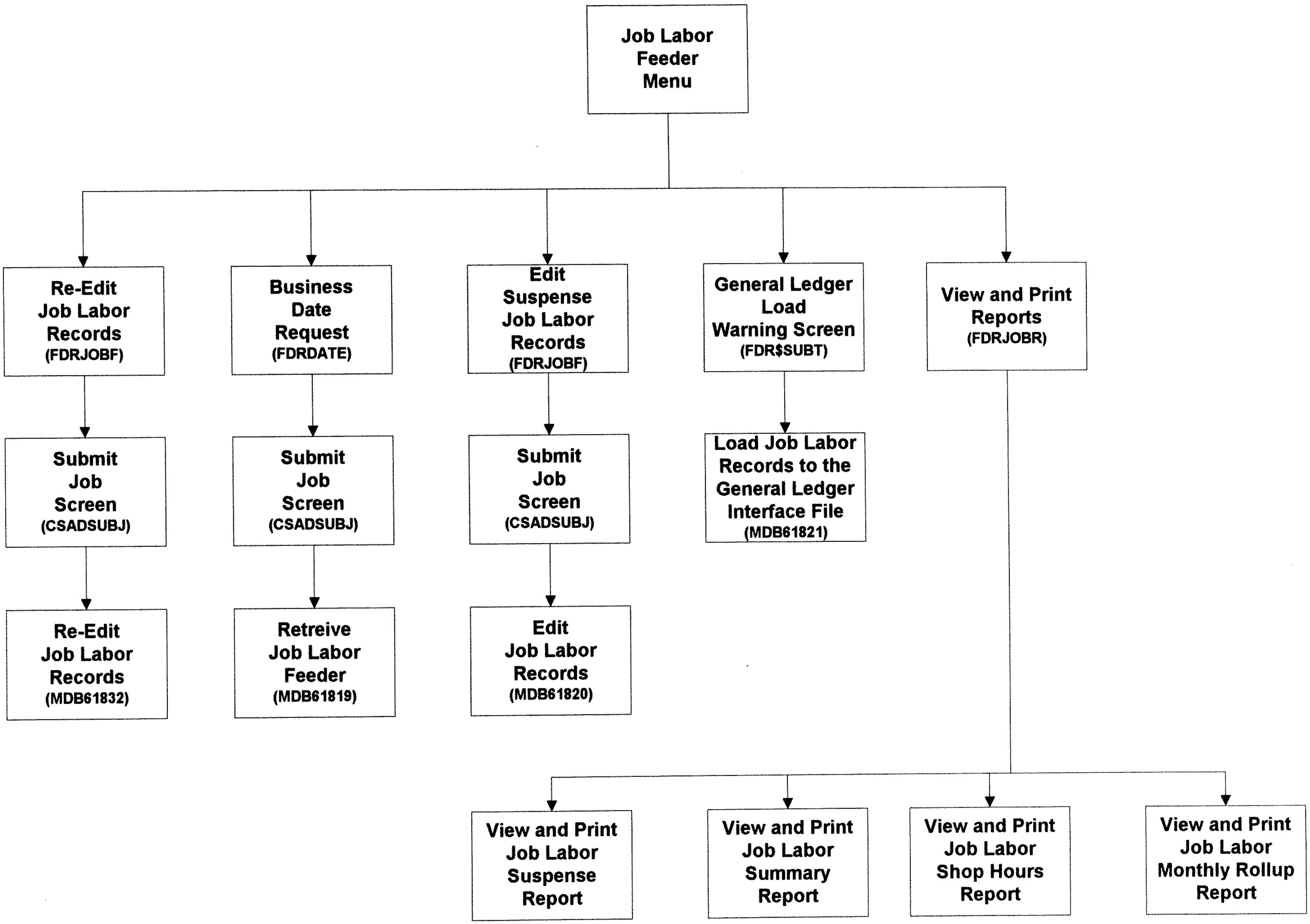
Middletown Works Feeders System



Salary Feeder Menu
Business Date Request (FDRDATE)
Edit Suspense Salary Records (FDRSLRYF)
General Ledger Load Warning Screen (FDR$SUBT)
View and Print Reports (FDRSLRYR)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
View and Print Salary Suspense Report
View and Print Salary Listing Report
Retrieve Salary Feeder (MDB61816)
Edited Salary Feeder (MDB61817)
Load Salary Records to the General Ledger Interface File (MDB61818)

Hierarchy Chart

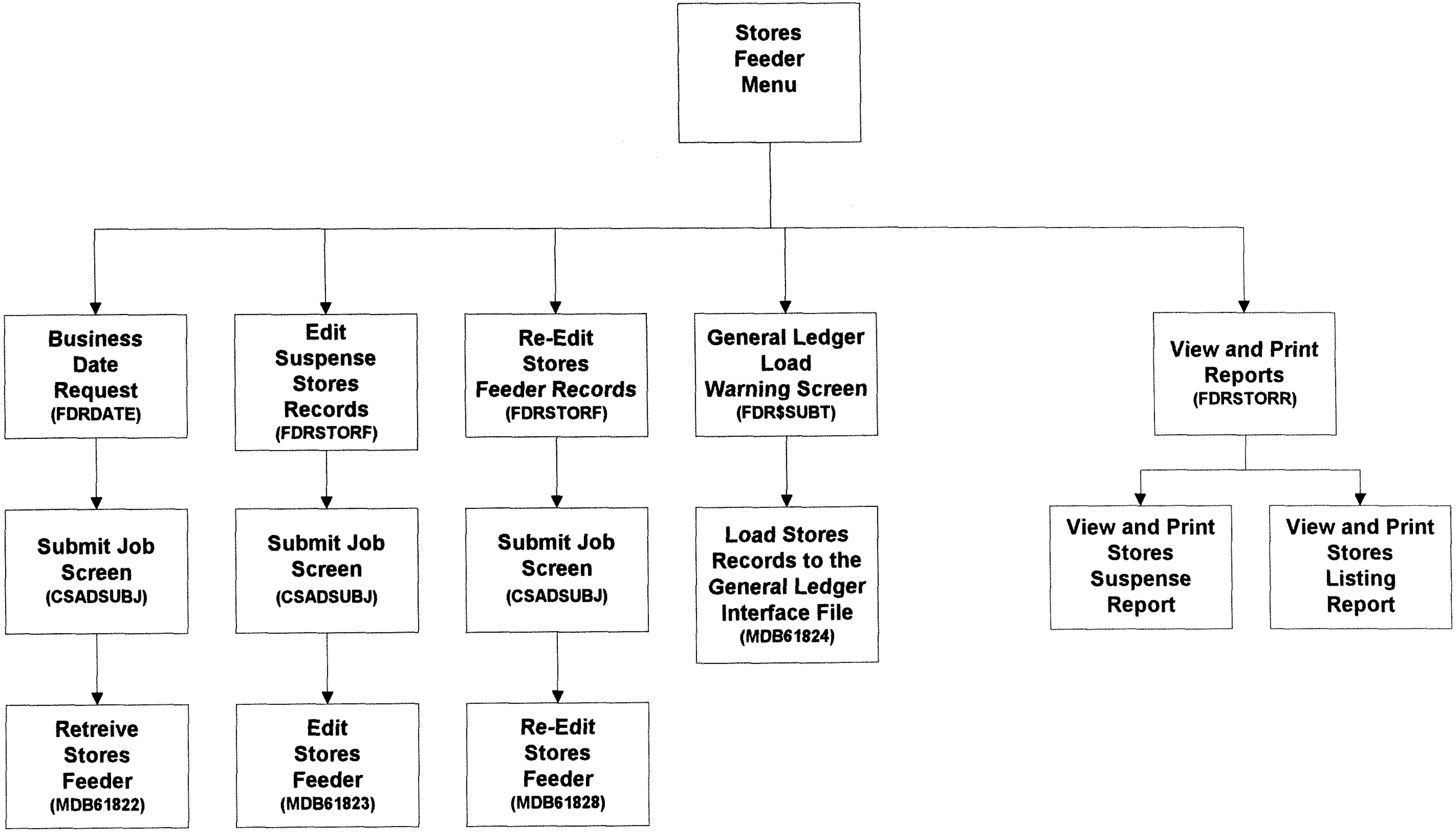
Middletown Works Feeders System



Job Labor Feeder Menu
Re-Edit Job Labor Records (FDRJOBF)
Business Date Request (FDRDATE)
Edit Suspense Job Labor Records (FDRJOBF)
General Ledger Load Warning Screen (FDR$SUBT)
View and Print Reports (FDRJOBR)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
Load Job Labor Records to the General Ledger Interface File (MDB61821)
Re-Edit Job Labor Records (MDB61832)
Retreive Job Labor Feeder (MDB61819)
Edit Job Labor Records (MDB61820)
View and Print Job Labor Suspense Report
View and Print Job Labor Summary Report
View and Print Job Labor Shop Hours Report
View and Print Job Labor Monthly Rollup Report

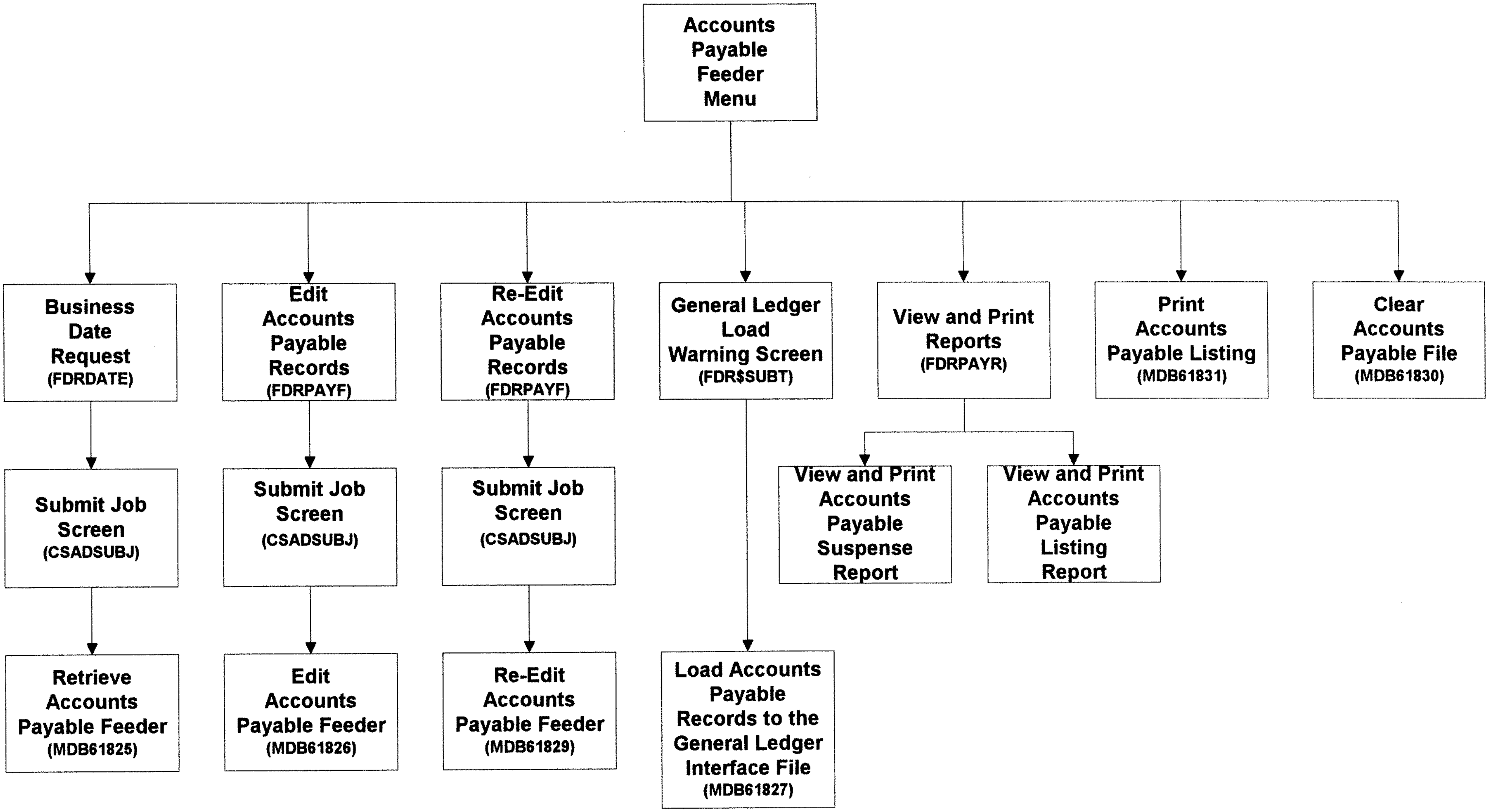
Hierarchy Chart

Middletown Works Feeders System


Stores Feeder Menu
Business Date Request (FDRDATE)
Edit Suspense Stores Records (FDRSTORF)
Re-Edit Stores Feeder Records (FDRSTORF)
General Ledger Load Warning Screen (FDR$SUBT)
View and Print Reports (FDRSTORR)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
Load Stores Records to the General Ledger Interface File (MDB61824)
View and Print Stores Suspense Report
View and Print Stores Listing Report
Retreive Stores Feeder (MDB61822)
Edit Stores Feeder (MDB61823)
Re-Edit Stores Feeder (MDB61828)

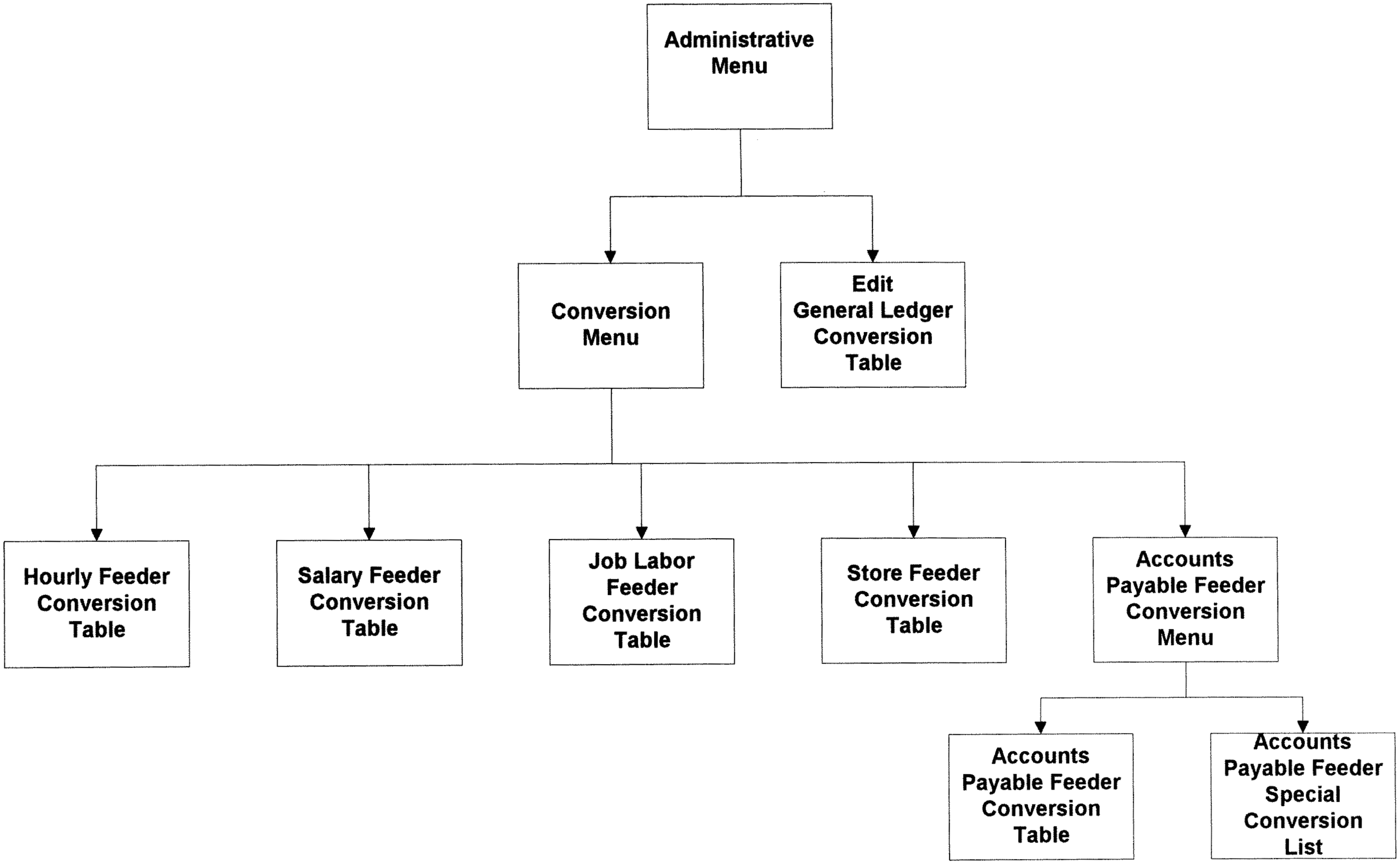
Hierarchy Chart

Middletown Works Feeders System


Accounts Payable Feeder Menu
Business Date Request (FDRDATE)
Edit Accounts Payable Records (FDRPAYF)
Re-Edit Accounts Payable Records (FDRPAYF)
General Ledger Load Warning Screen (FDR$SUBT)
View and Print Reports (FDRPAYR)
Print Accounts Payable Listing (MDB61831)
Clear Accounts Payable File (MDB61830)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
Submit Job Screen (CSADSUBJ)
View and Print Accounts Payable Suspense Report
View and Print Accounts Payable Listing Report
Retrieve Accounts Payable Feeder (MDB61825)
Edit Accounts Payable Feeder (MDB61826)
Re-Edit Accounts Payable Feeder (MDB61829)
Load Accounts Payable Records to the General Ledger Interface File (MDB61827)

Hierarchy Chart

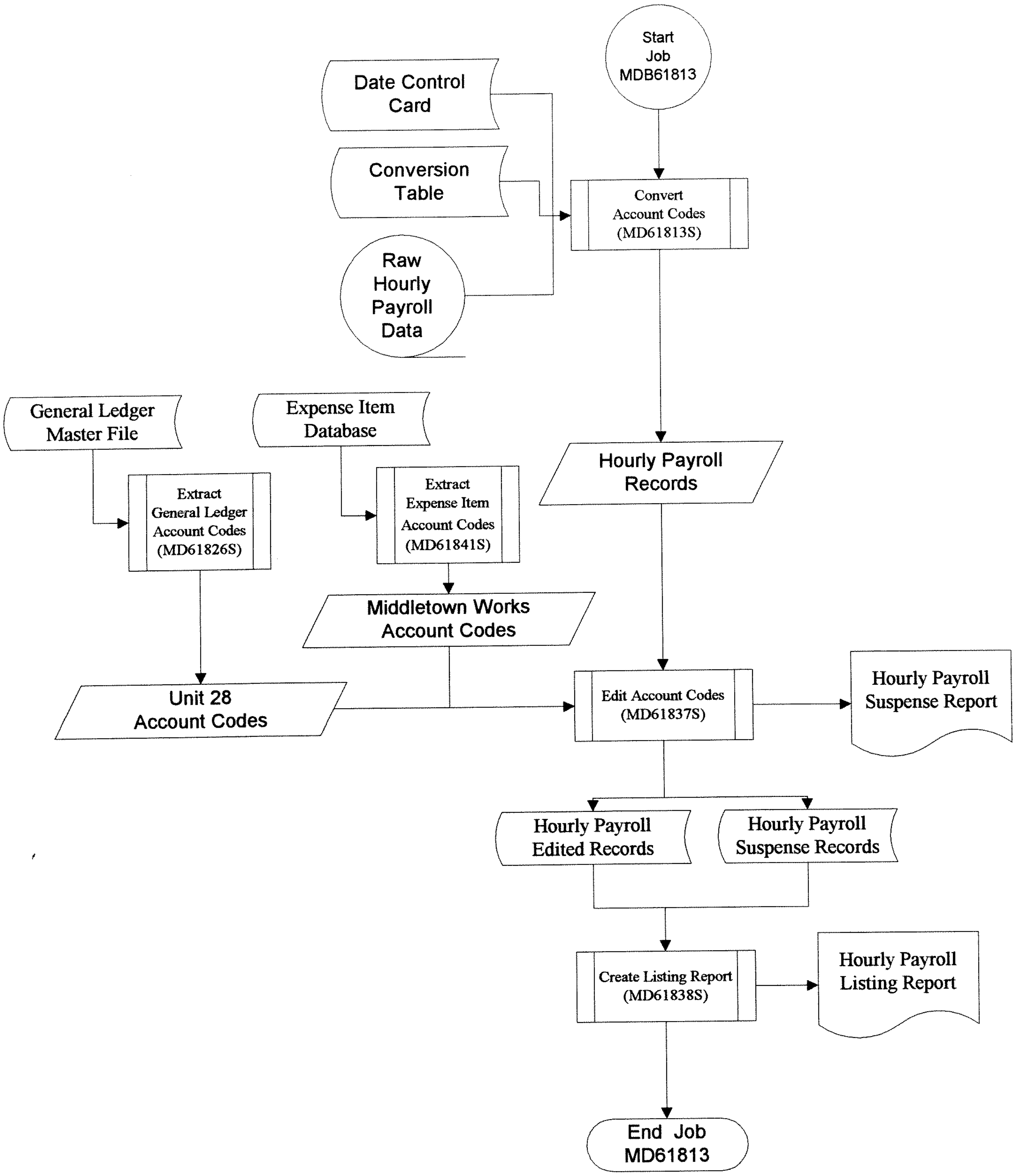
Middletown Works Feeders System



Administrative Menu
Conversion Menu
Edit General Ledger Conversion Table
Hourly Feeder Conversion Table
Salary Feeder Conversion Table
Job Labor Feeder Conversion Table
Store Feeder Conversion Table
Accounts Payable Feeder Conversion Menu
Accounts Payable Feeder Conversion Table
Accounts Payable Feeder Special Conversion List

Structured Flowchart

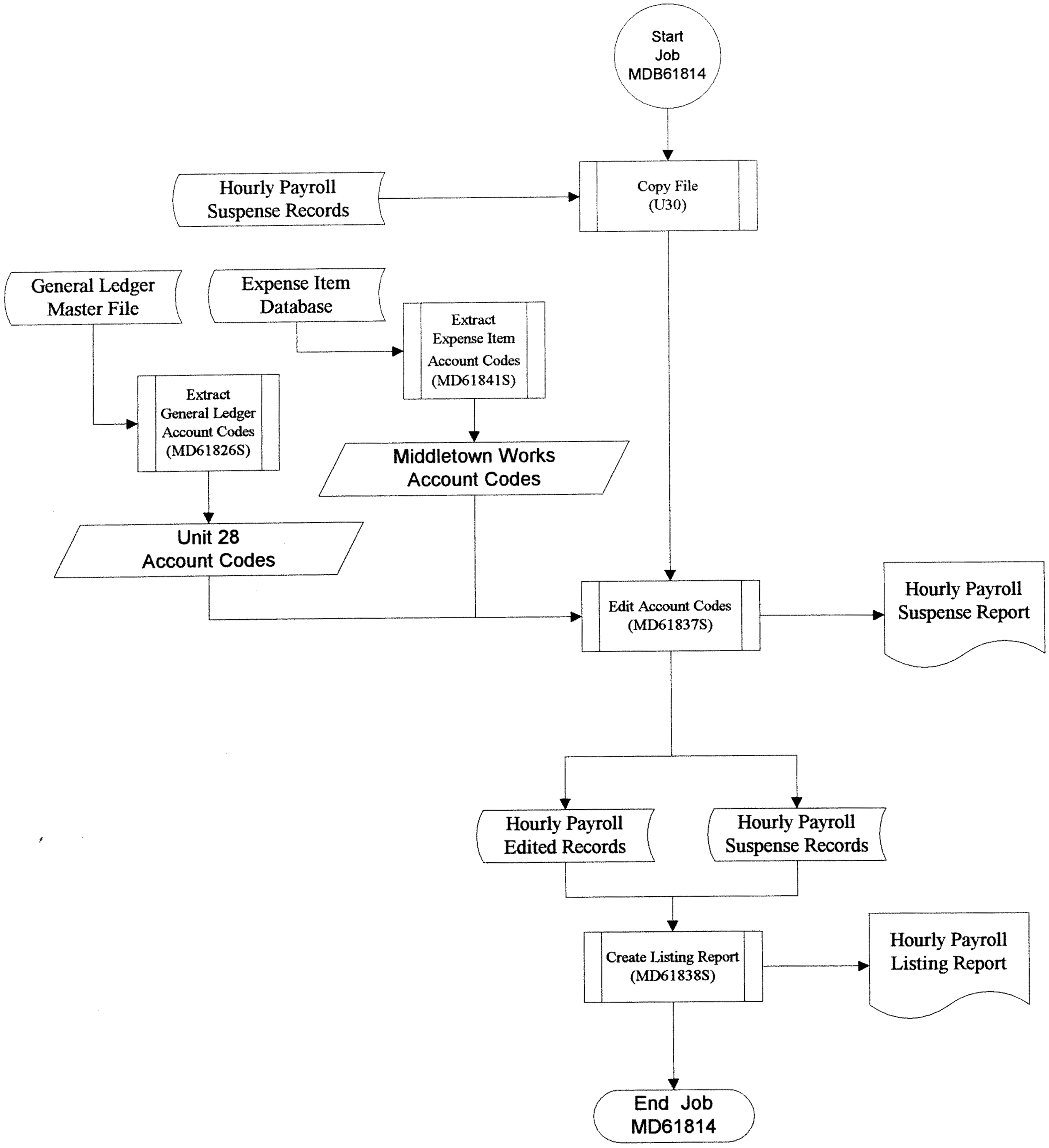
Middletown Works Feeders System
Retrieve Hourly Payroll Job(MDB61813)


Start Job MDB61813
Date Control Card
Conversion Table
Raw Hourly Payroll Data
Convert Account Codes (MD61813S)
Hourly Payroll Records
General Ledger Master File
Expense Item Database
Extract General Ledger Account Codes (MD61826S)
Extract Expense Item Account Codes (MD61841S)
Middletown Works Account Codes
Unit 28 Account Codes
Edit Account Codes (MD61837S)
Hourly Payroll Suspense Report
Hourly Payroll Edited Records
Hourly Payroll Suspense Records
Create Listing Report (MD61838S)
Hourly Payroll Listing Report
End Job MD61813

Structured Flowchart

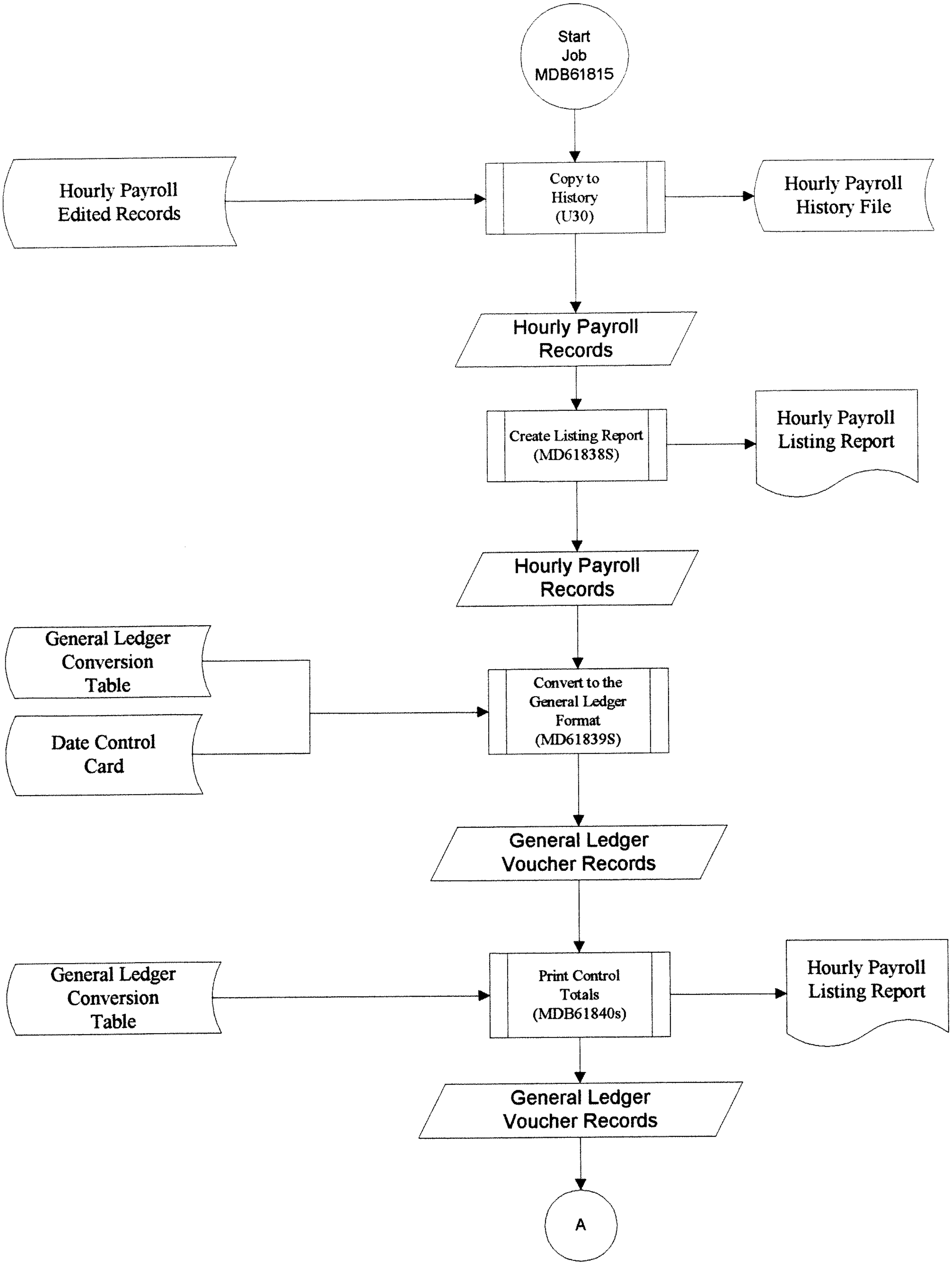
Middletown Works Feeders System

Edit Hourly Payroll Job (MDB61814)


Start Job MDB61814
Hourly Payroll Suspense Records
Copy File (U30)
General Ledger Master File
Expense Item Database
Extract Expense Item Account Codes (MD61841S)
Extract General Ledger Account Codes (MD61826S)
Middletown Works Account Codes
Unit 28 Account Codes
Edit Account Codes (MD61837S)
Hourly Payroll Suspense Report
Hourly Payroll Edited Records
Hourly Payroll Suspense Records
Create Listing Report (MD61838S)
Hourly Payroll Listing Report
End Job MD61814

Structured Flowchart

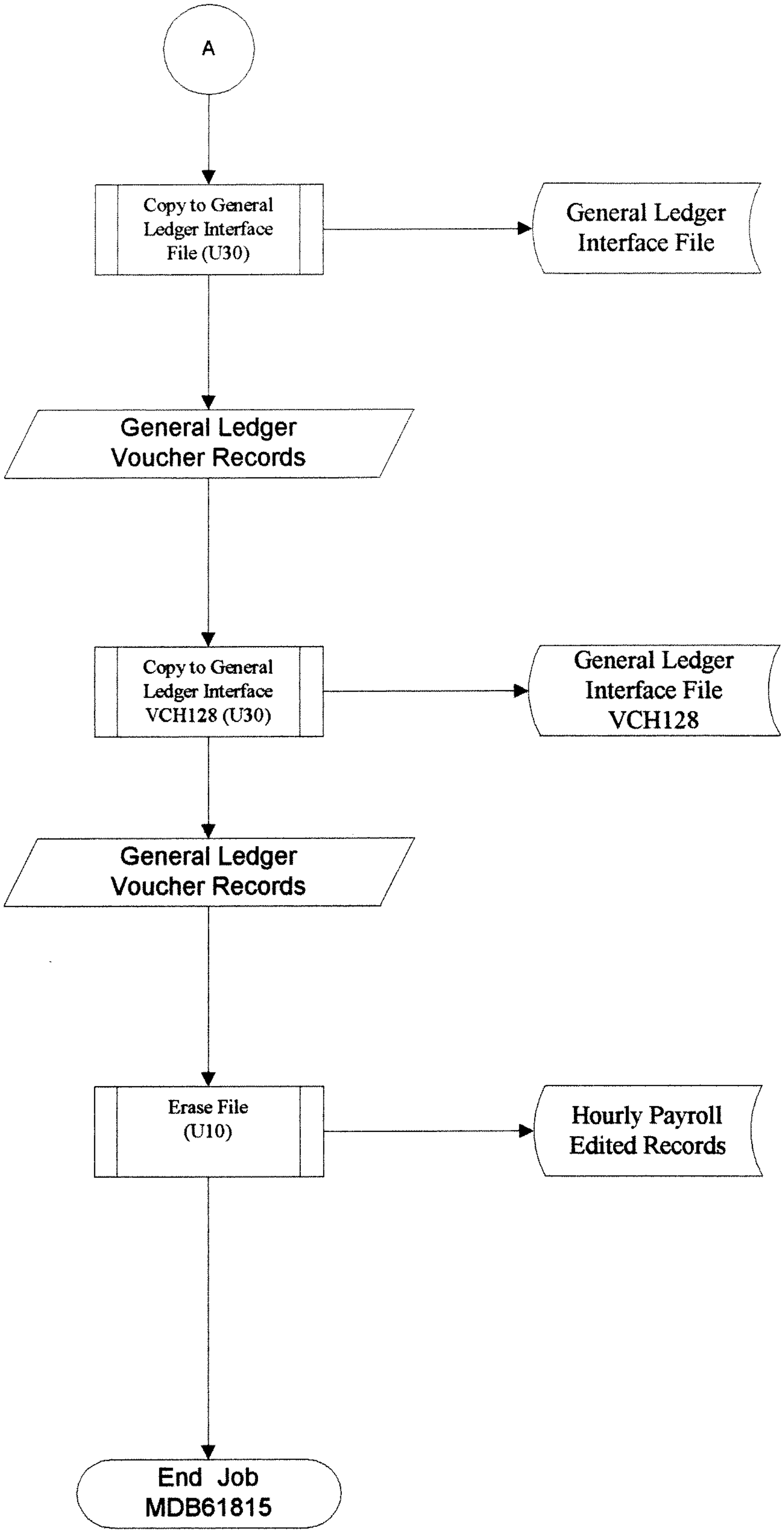
Middletown Works Feeders System
Load Hourly Payroll Job (MDB61815)


Start Job MDB61815
Hourly Payroll Edited Records
Copy to History (U30)
Hourly Payroll History File
Hourly Payroll Records
Create Listing Report (MD61838S)
Hourly Payroll Listing Report
Hourly Payroll Records
General Ledger Conversion Table
Date Control Card
Convert to the General Ledger Format (MD61839S)
General Ledger Voucher Records
General Ledger Conversion Table
Print Control Totals (MDB61840s)
Hourly Payroll Listing Report
General Ledger Voucher Records
A

Structured Flowchart

Middletown Works Feeders System

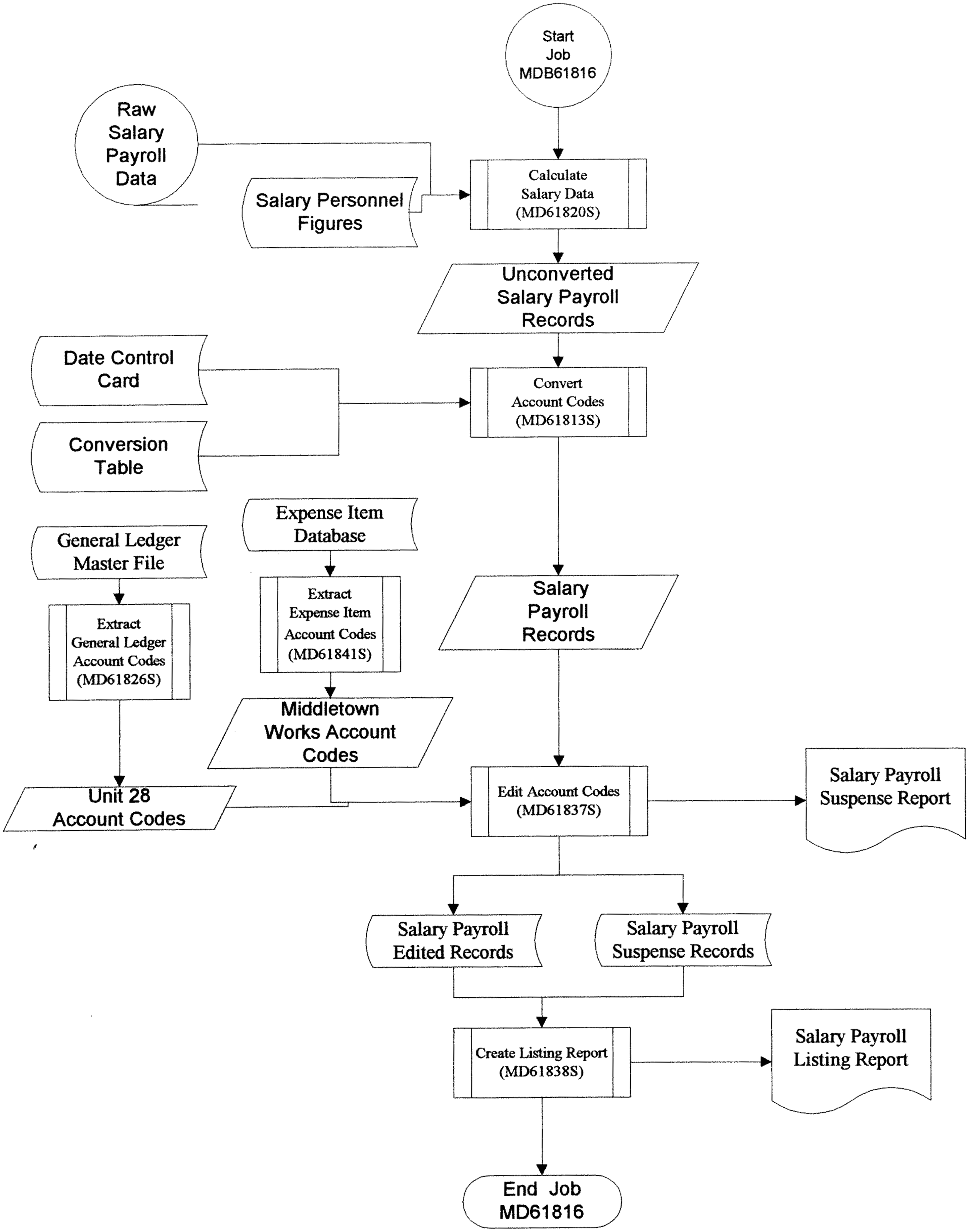
Load Hourly Payroll Job (MDB61815)


A
Copy to General Ledger Interface File (U30)
General Ledger Interface File
General Ledger Voucher Records
Copy to General Ledger Interface VCH128 (U30)
General Ledger Interface File VCH128
General Ledger Voucher Records
Erase File (U10)
Hourly Payroll Edited Records
End Job MDB61815

Structured Flowchart

Middletown Works Feeders System

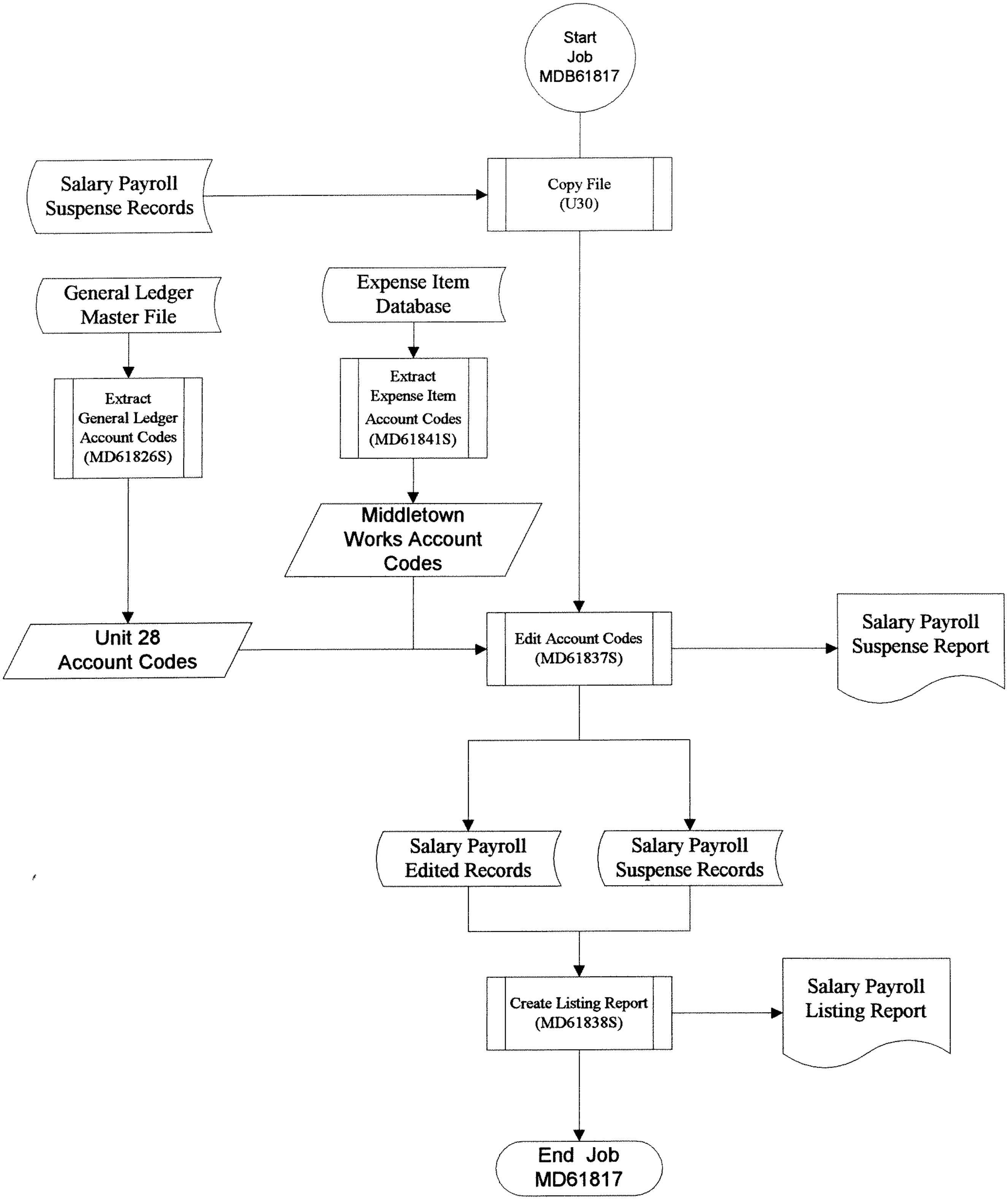
Retrieve Salary Payroll Job(MDB61816)


Start Job MDB61816
Raw Salary Payroll Data
Salary Personnel Figures
Calculate Salary Data (MD61820S)
Unconverted Salary Payroll Records
Date Control Card
Conversion Table
Convert Account Codes (MD61813S)
General Ledger Master File
Expense Item Database
Extract General Ledger Account Codes (MD61826S)
Extract Expense Item Account Codes (MD61841S)
Salary Payroll Records
Middletown Works Account Codes
Unit 28 Account Codes
Edit Account Codes (MD61837S)
Salary Payroll Suspense Report
Salary Payroll Edited Records
Salary Payroll Suspense Records
Create Listing Report (MD61838S)
Salary Payroll Listing Report
End Job MD61816

Structured Flowchart

Middletown Works Feeders System

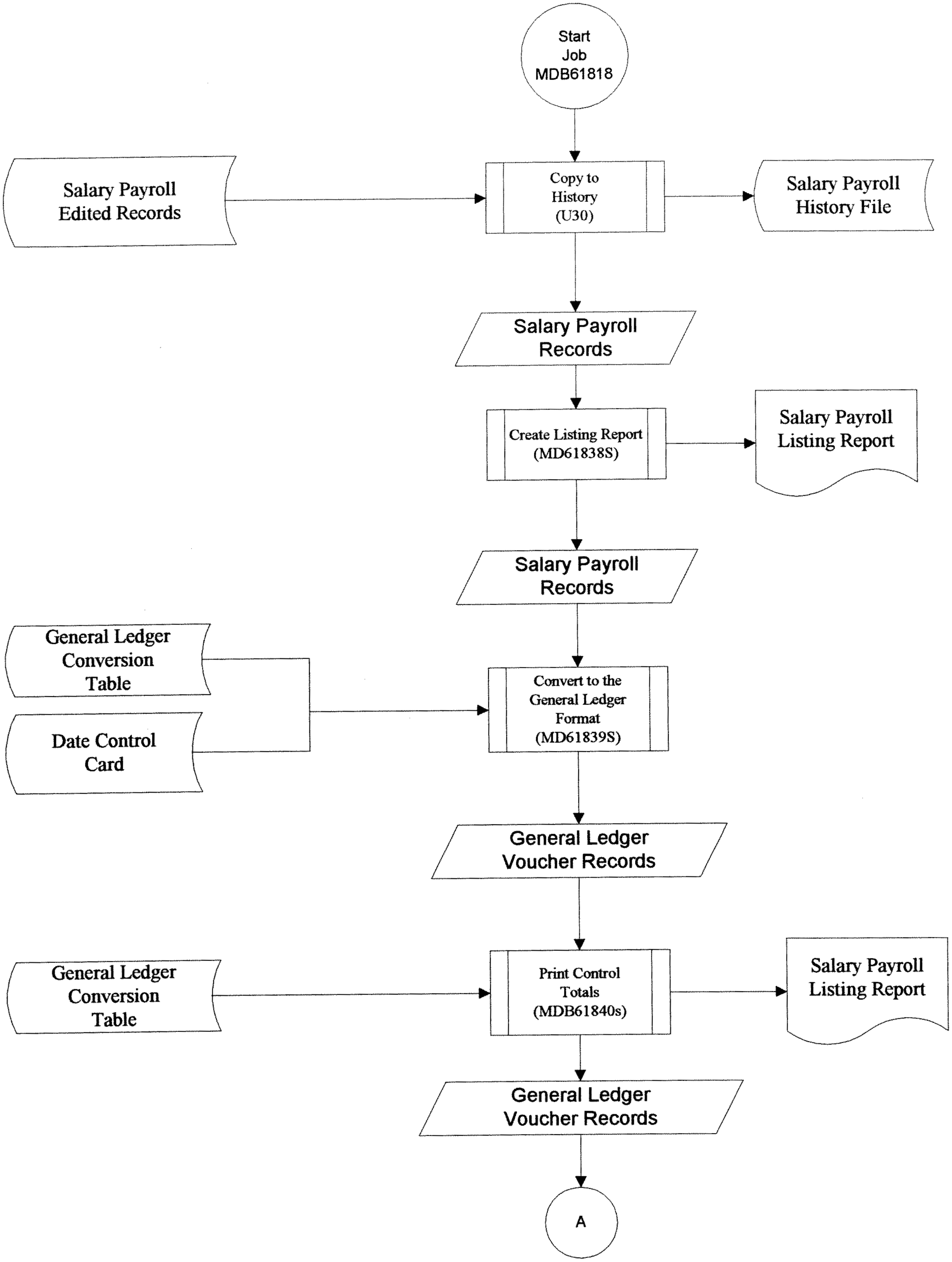
Edit Salary Payroll Job (MDB61817)


Start Job MDB61817
Salary Payroll Suspense Records
Copy File (U30)
General Ledger Master File
Expense Item Database
Extract General Ledger Account Codes (MD61826S)
Extract Expense Item Account Codes (MD61841S)
Middletown Works Account Codes
Unit 28 Account Codes
Edit Account Codes (MD61837S)
Salary Payroll Suspense Report
Salary Payroll Edited Records
Salary Payroll Suspense Records
Create Listing Report (MD61838S)
Salary Payroll Listing Report
End Job MD61817

Structured Chart

Middletown Works Feeders System

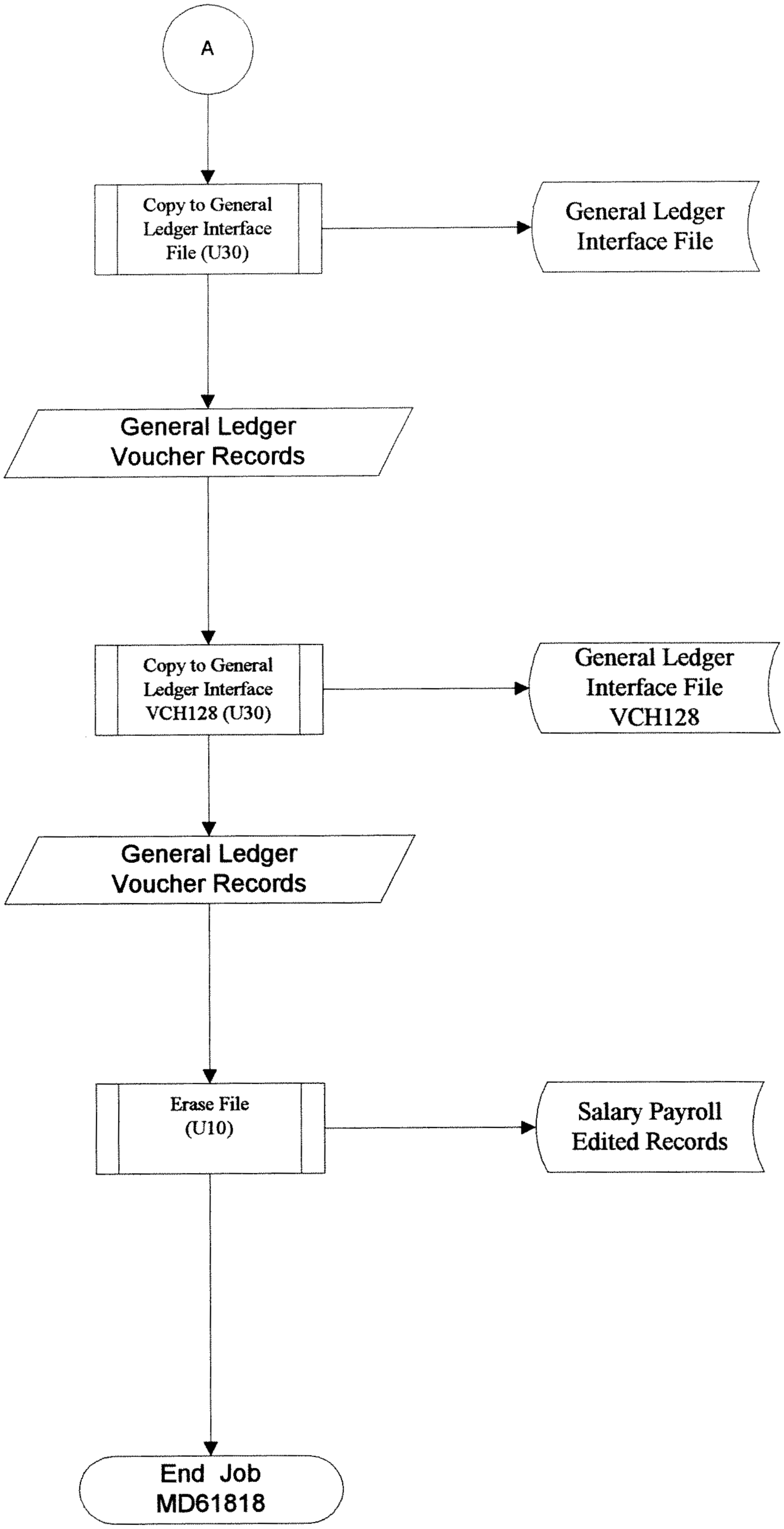
Load Salary Payroll Job (MDB61818)


Start Job MDB61818
Salary Payroll Edited Records
Copy to History (U30)
Salary Payroll History File
Salary Payroll Records
Create Listing Report (MD61838S)
Salary Payroll Listing Report
Salary Payroll Records
General Ledger Conversion Table
Date Control Card
Convert to the General Ledger Format (MD61839S)
General Ledger Voucher Records
General Ledger Conversion Table
Print Control Totals (MDB61840s)
Salary Payroll Listing Report
General Ledger Voucher Records
A

Structured Chart

Middletown Works Feeders System

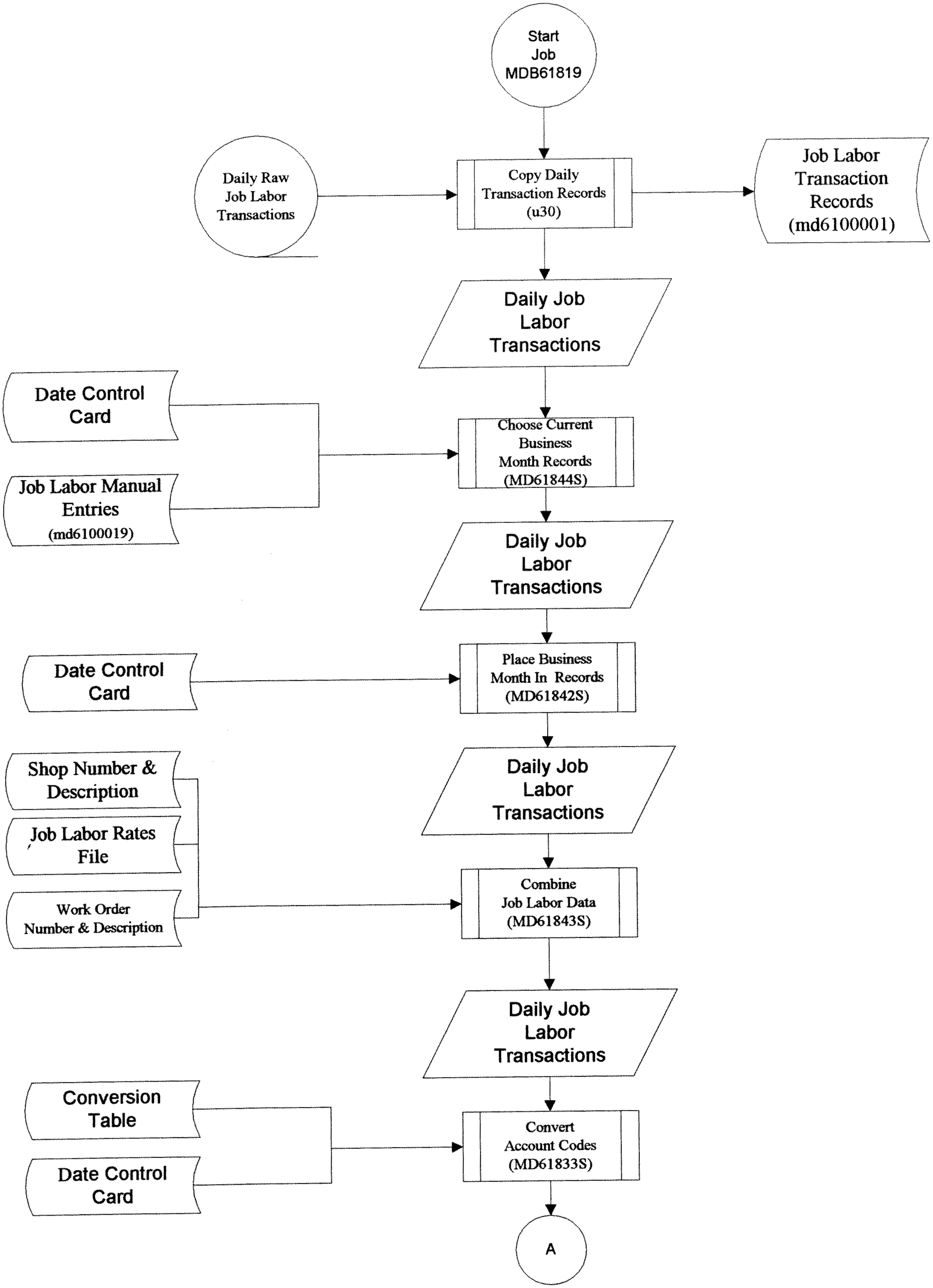
Load Salary Payroll Job (MDB61818)


A
Copy to General Ledger Interface File (U30)
General Ledger Interface File
General Ledger Voucher Records
Copy to General Ledger Interface VCH128 (U30)
General Ledger Interface File VCH128
General Ledger Voucher Records
Erase File (U10)
Salary Payroll Edited Records
End Job MD61818

Structured Chart

Middletown Works Feeders System

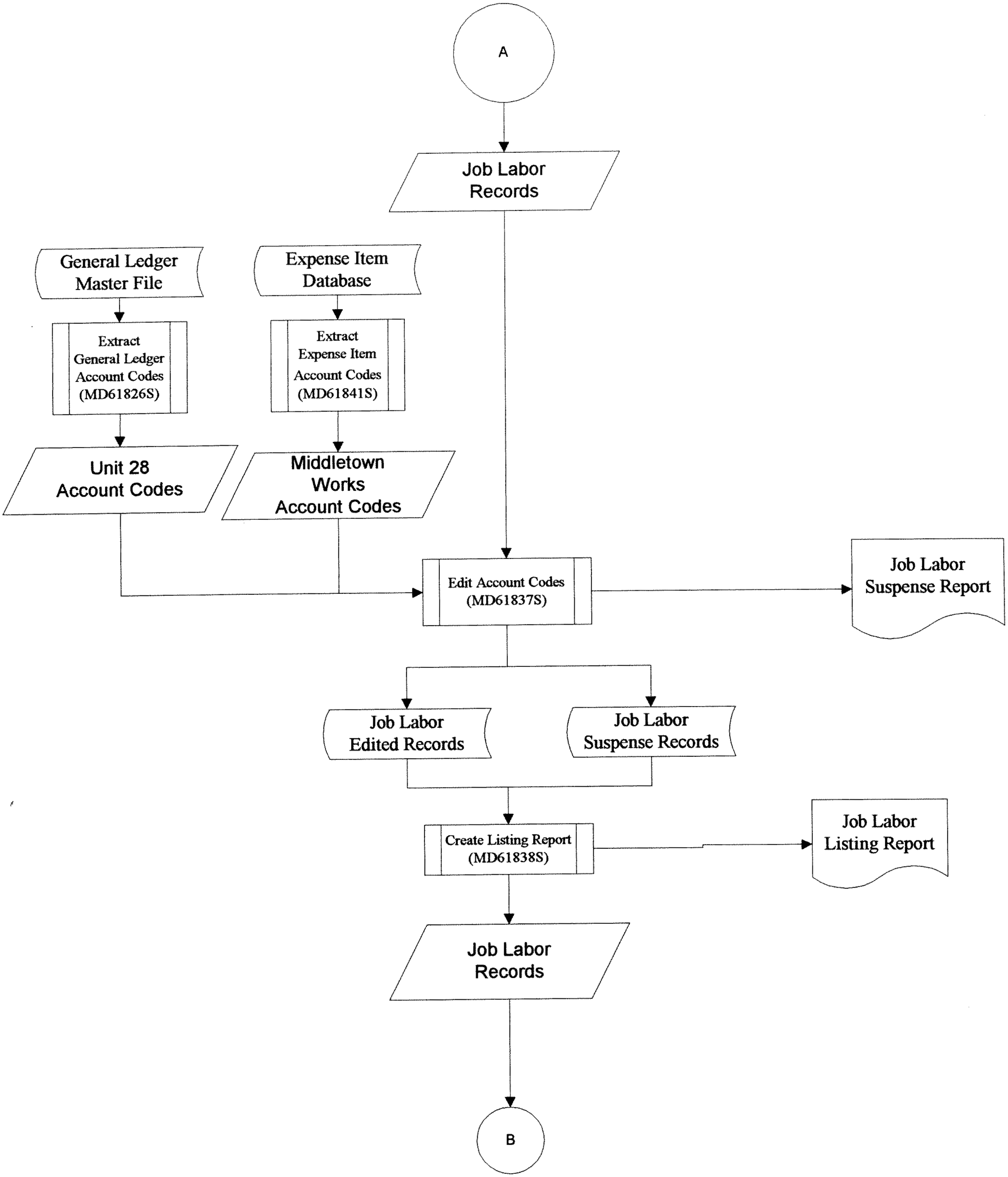
Retrieve Job Labor Job(MDB61819)


Start Job MDB61819
Daily Raw Job Labor Transactions
Copy Daily Transaction Records (u30)
Job Labor Transaction Records (md6100001)
Daily Job Labor Transactions
Date Control Card
Job Labor Manual Entries (md6100019)
Choose Current Business Month Records (MD61844S)
Daily Job Labor Transactions
Date Control Card
Place Business Month In Records (MD61842S)
Shop Number & Description
Job Labor Rates File
Work Order Number & Description
Daily Job Labor Transactions
Combine Job Labor Data (MD61843S)
Daily Job Labor Transactions
Conversion Table
Date Control Card
Convert Account Codes (MD61833S)
A

Structured Chart

Middletown Works Feeders System

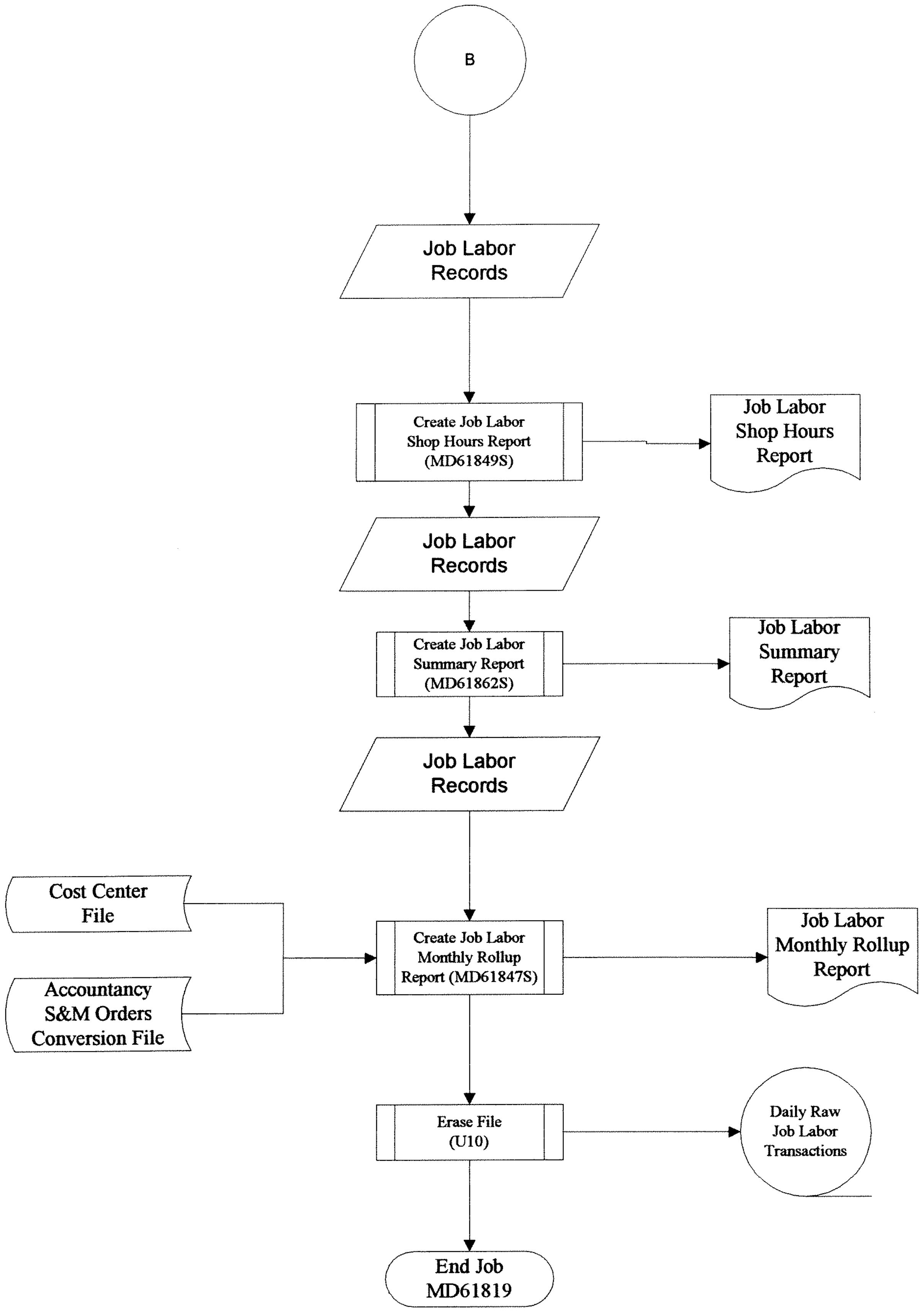
Retrieve Job Labor Job(MDB61819)


A
Job Labor Records
General Ledger Master File
Expense Item Database
Extract General Ledger Account Codes (MD61826S)
Extract Expense Item Account Codes (MD61841S)
Unit 28 Account Codes
Middletown Works Account Codes
Edit Account Codes (MD61837S)
Job Labor Suspense Report
Job Labor Edited Records
Job Labor Suspense Records
Create Listing Report (MD61838S)
Job Labor Listing Report
Job Labor Records
B

Structured Chart

Middletown Works Feeders System

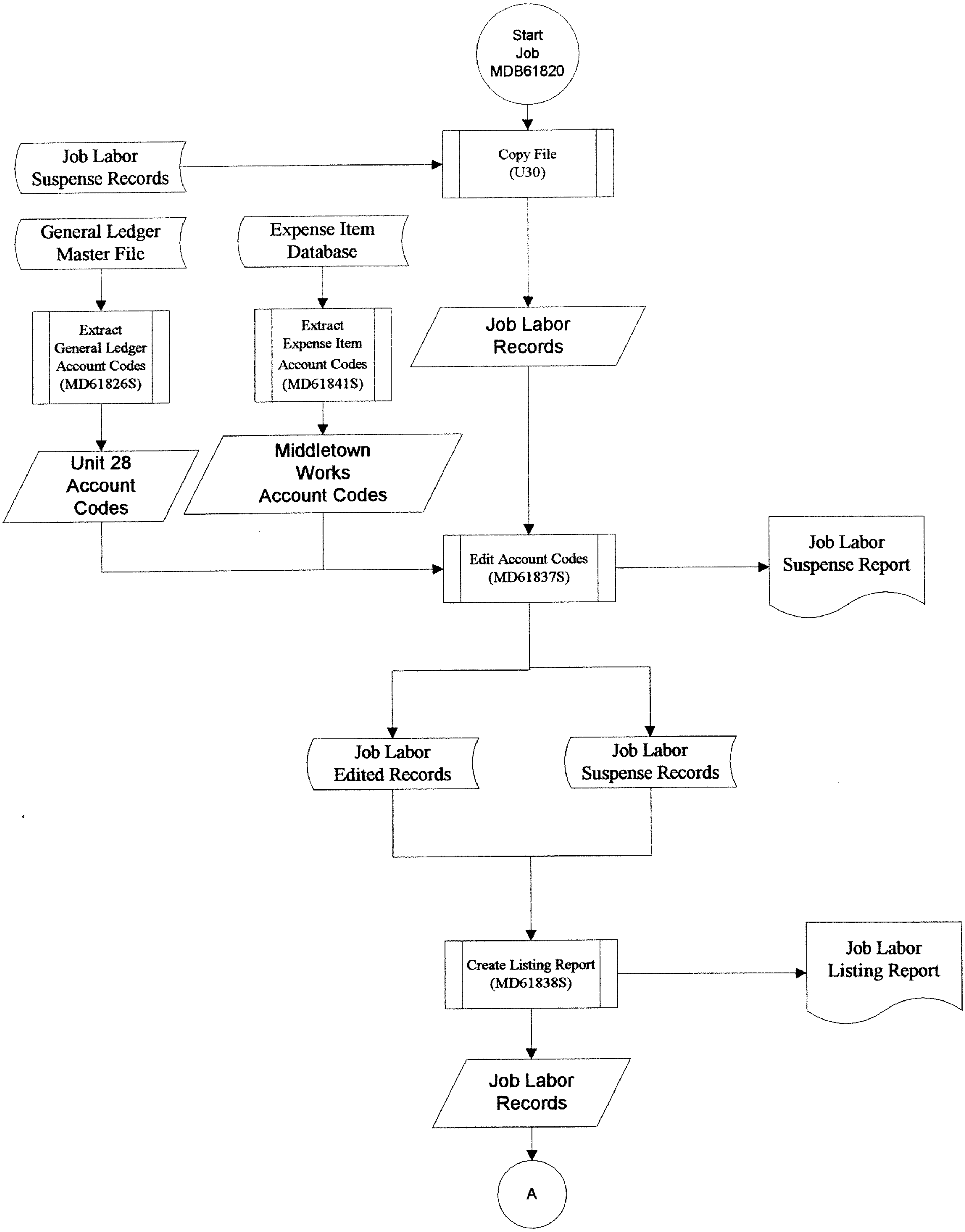
Retrieve Job Labor Job(MDB61819)


B
Job Labor Records
Create Job Labor Shop Hours Report (MD61849S)
Job Labor Shop Hours Report
Job Labor Records
Create Job Labor Summary Report (MD61862S)
Job Labor Summary Report
Job Labor Records
Cost Center File
Accountancy S&M Orders Conversion File
Create Job Labor Monthly Rollup Report (MD61847S)
Job Labor Monthly Rollup Report
Erase File (U10)
Daily Raw Job Labor Transactions
End Job MD61819

Structured Chart

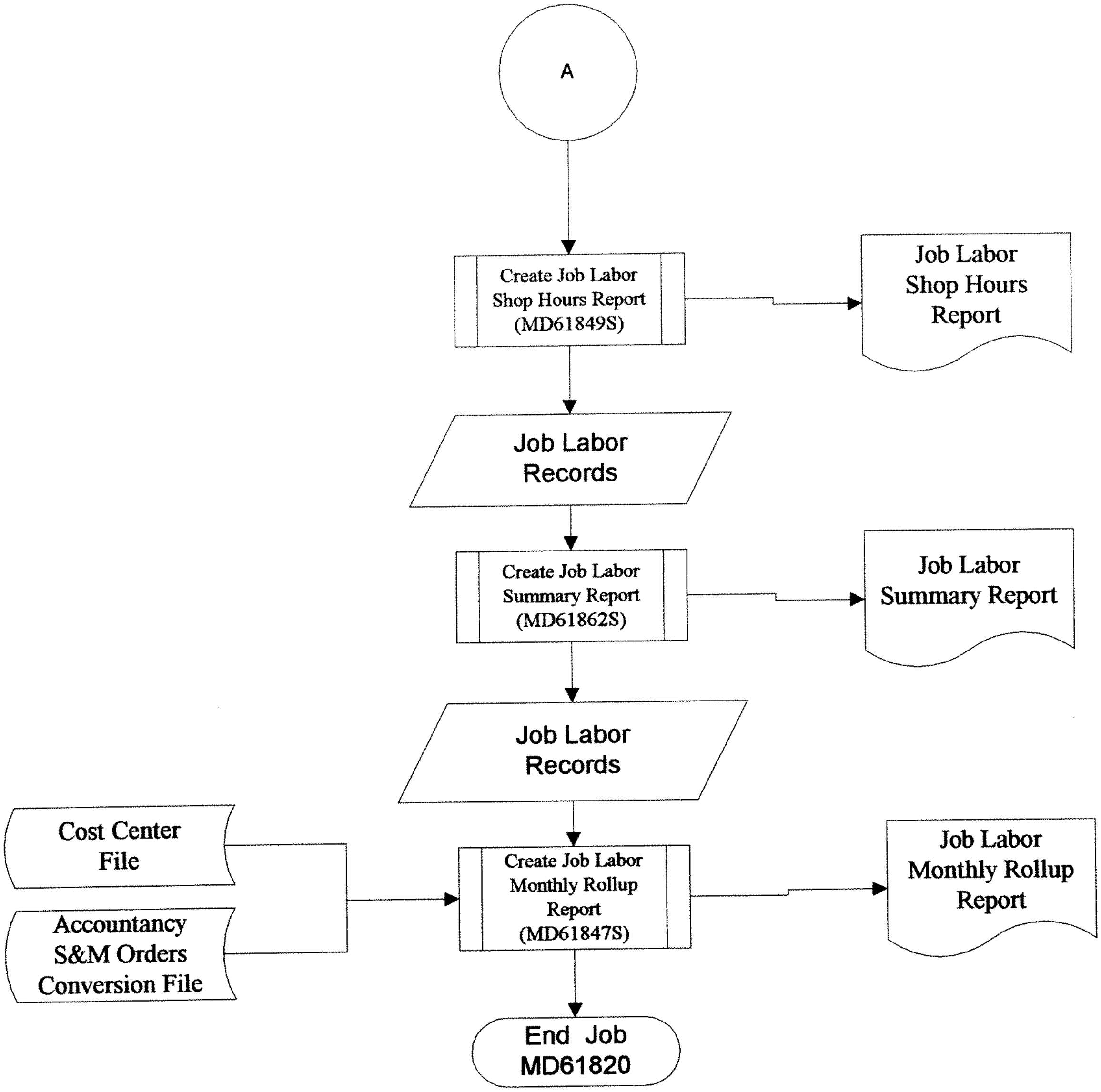
Middletown Works Feeders System

Edit Job Labor Job (MDB61820)


Start Job MDB61820
Copy File (U30)
Job Labor Suspense Records
General Ledger Master File
Expense Item Database
Extract General Ledger Account Codes (MD61826S)
Extract Expense Item Account Codes (MD61841S)
Job Labor Records
Unit 28 Account Codes
Middletown Works Account Codes
Edit Account Codes (MD61837S)
Job Labor Suspense Report
Job Labor Edited Records
Job Labor Suspense Records
Create Listing Report (MD61838S)
Job Labor Listing Report
Job Labor Records
A

Structured Chart

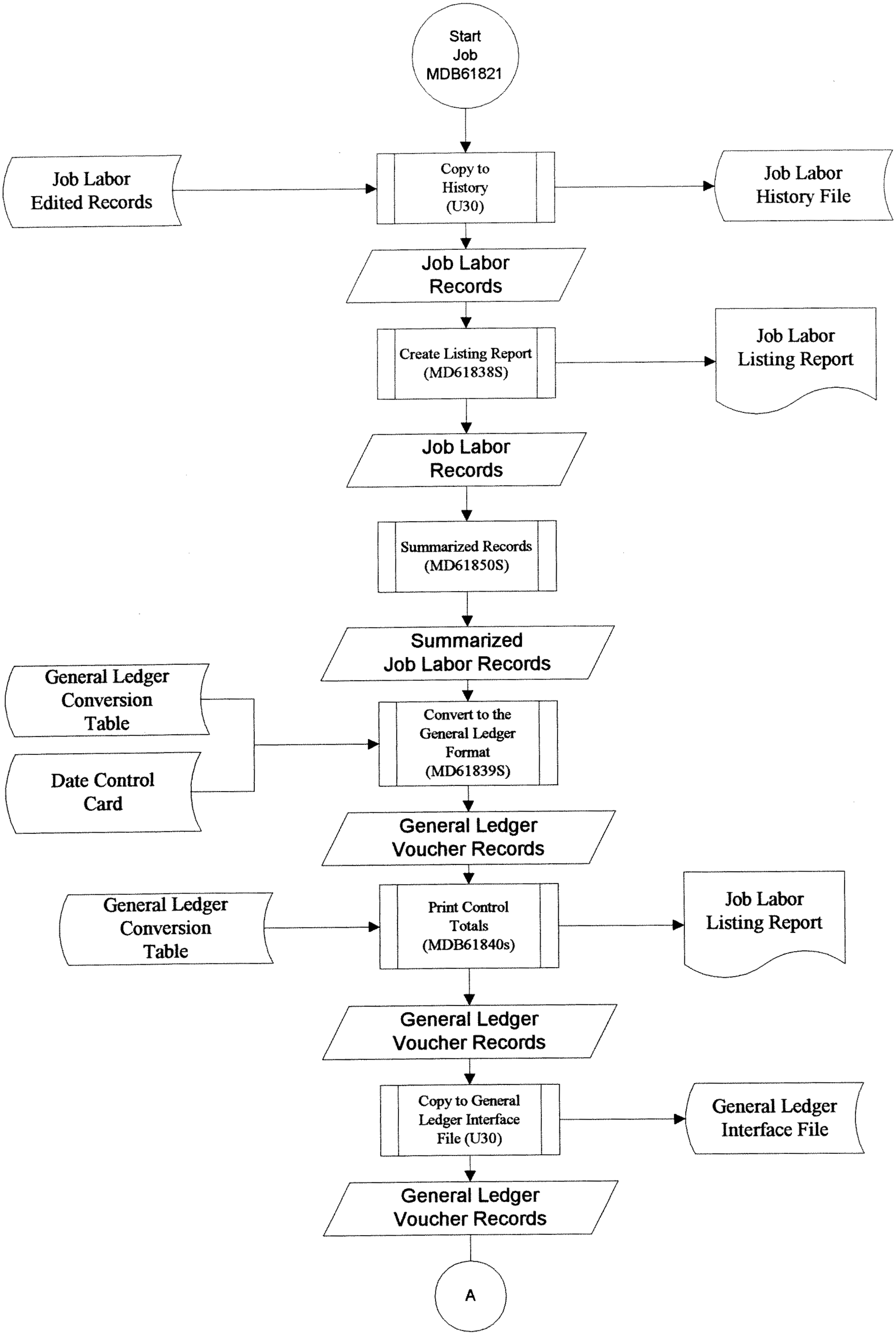
Middletown Works Feeders System
Edit Job Labor Job (MDB61820)


A
Create Job Labor Shop Hours Report (MD61849S)
Job Labor Shop Hours Report
Job Labor Records
Create Job Labor Summary Report (MD61862S)
Job Labor Summary Report
Job Labor Records
Cost Center File
Accountancy S&M Orders Conversion File
Create Job Labor Monthly Rollup Report (MD61847S)
Job Labor Monthly Rollup Report
End Job MD61820

Structured Chart

Middletown Works Feeders System

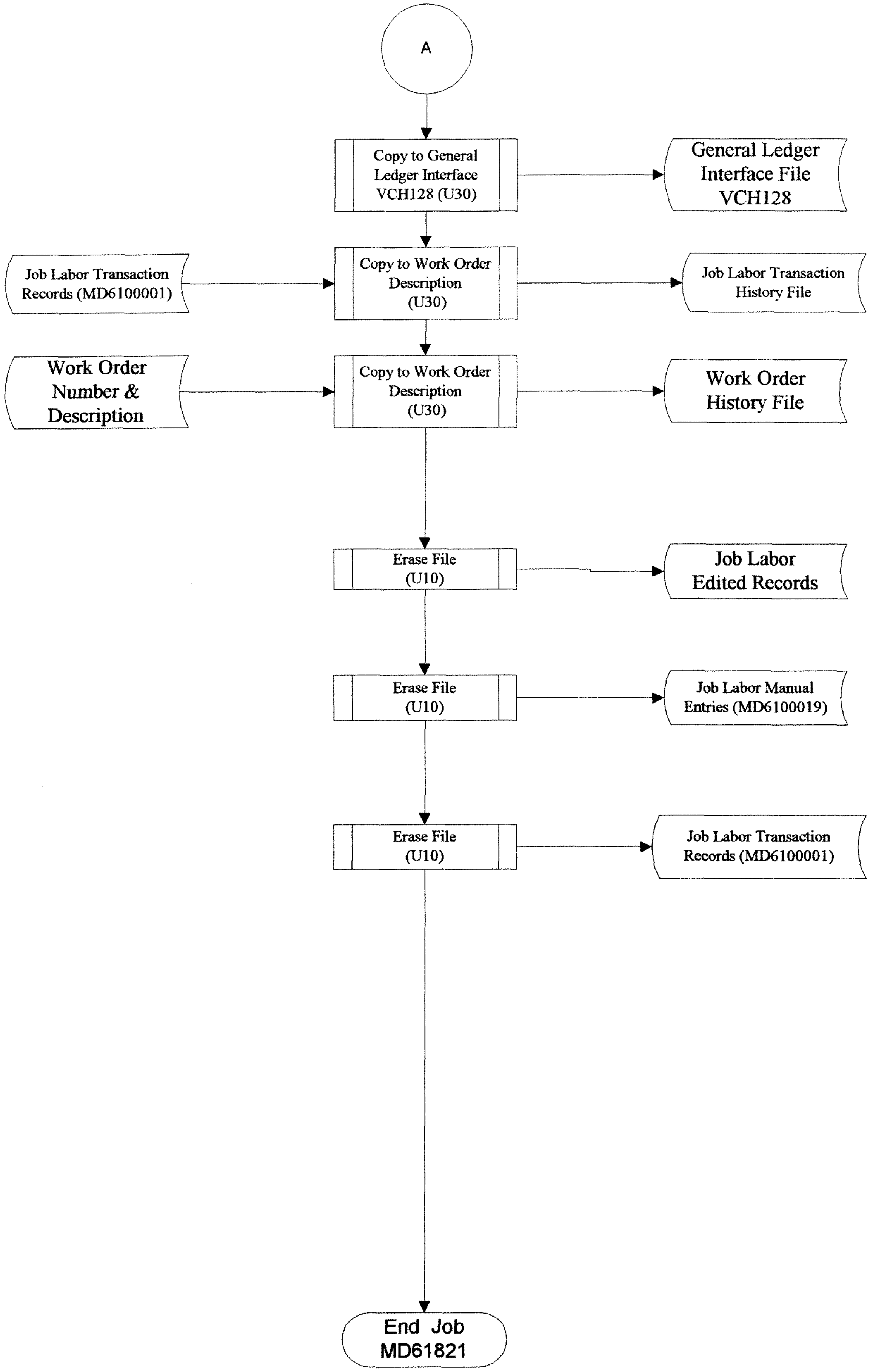
Load Job Labor Job (MDB61821)


Start Job MDB61821
Job Labor Edited Records
Copy to History (U30)
Job Labor History File
Job Labor Records
Create Listing Report (MD61838S)
Job Labor Listing Report
Job Labor Records
Summarized Records (MD61850S)
Summarized Job Labor Records
General Ledger Conversion Table
Date Control Card
Convert to the General Ledger Format (MD61839S)
General Ledger Voucher Records
General Ledger Conversion Table
Print Control Totals (MDB61840s)
Job Labor Listing Report
General Ledger Voucher Records
Copy to General Ledger Interface File (U30)
General Ledger Interface File
General Ledger Voucher Records
A

Structured Chart

Middletown Works Feeders System

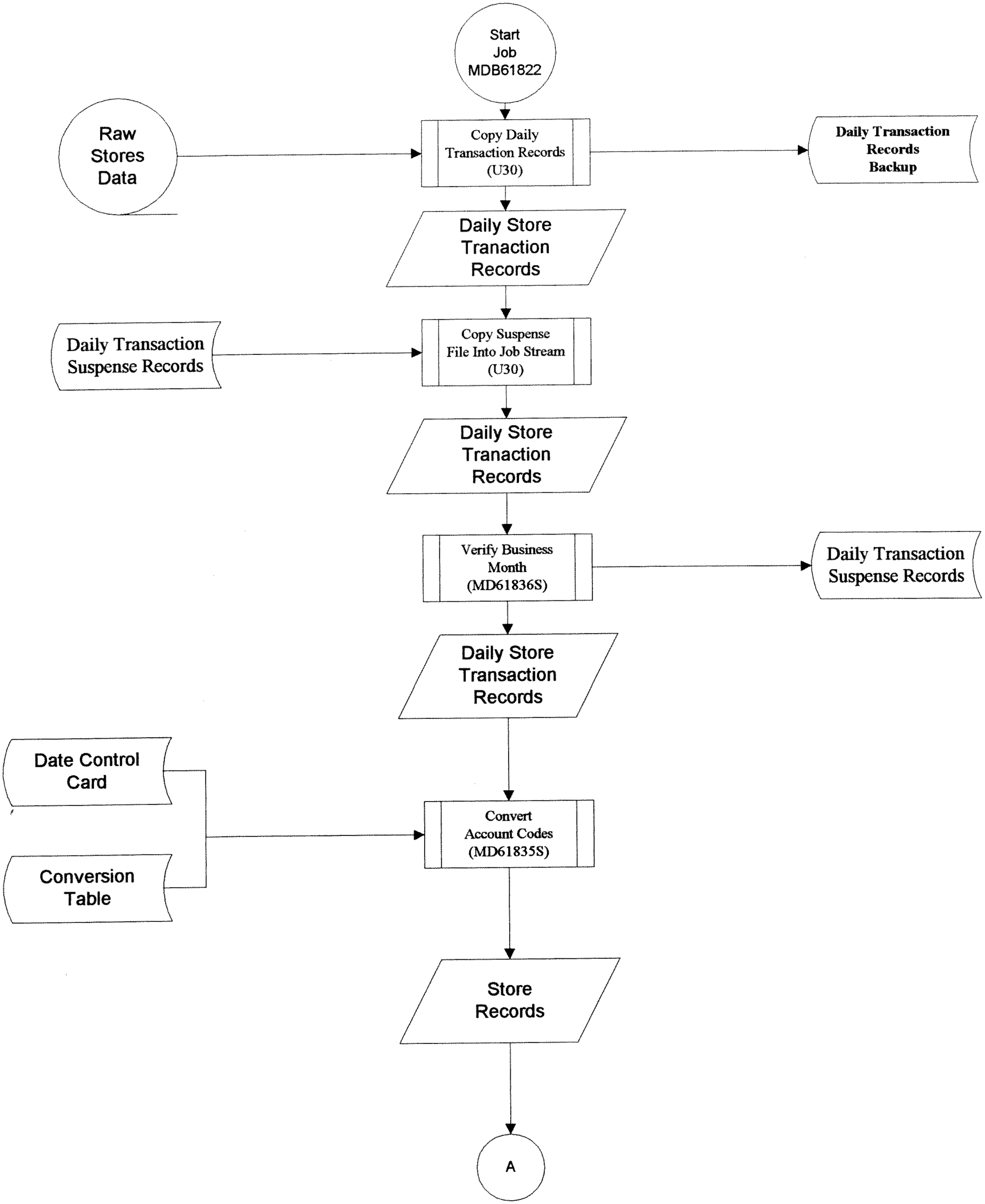
Load Job Labor Job (MDB61821)


A
Copy to General Ledger Interface VCH128 (U30)
General Ledger Interface File VCH128
Job Labor Transaction Records (MD6100001)
Copy to Work Order Description (U30)
Job Labor Transaction History File
Work Order Number & Description
Copy to Work Order Description (U30)
Work Order History File
Erase File (U10)
Job Labor Edited Records
Erase File (U10)
Job Labor Manual Entries (MD6100019)
Erase File (U10)
Job Labor Transaction Records (MD6100001)
End Job MD61821

Structured Chart

Middletown Works Feeders System

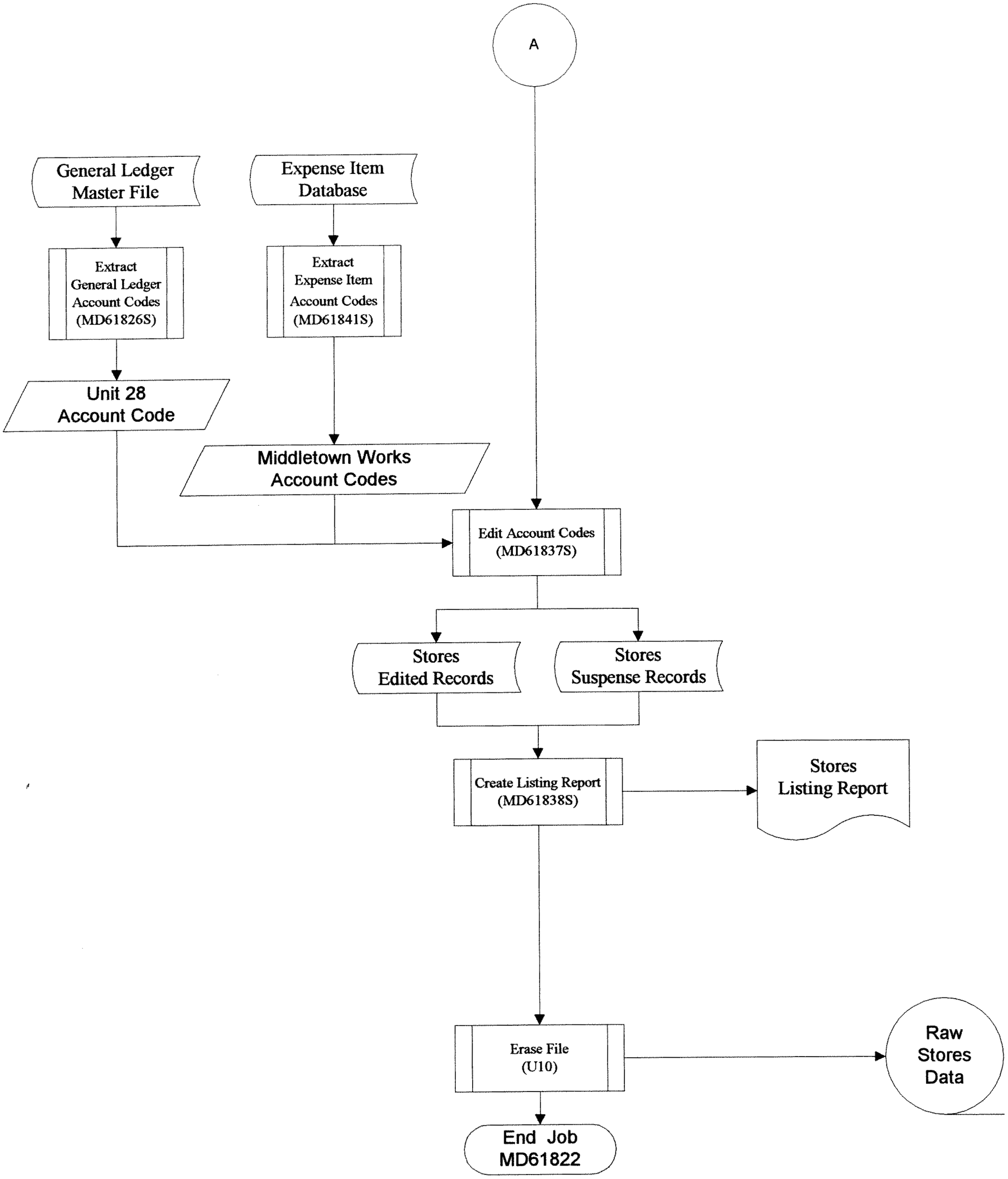
Retrieve Stores Job(MDB61822)


Start Job MDB61822
Raw Stores Data
Copy Daily Transaction Records (U30)
Daily Transaction Records Backup
Daily Store Tranaction Records
Daily Transaction Suspense Records
Copy Suspense File Into Job Stream (U30)
Daily Store Tranaction Records
Verify Business Month (MD61836S)
Daily Transaction Suspense Records
Daily Store Transaction Records
Date Control Card
Conversion Table
Convert Account Codes (MD61835S)
Store Records
A

Structured Chart

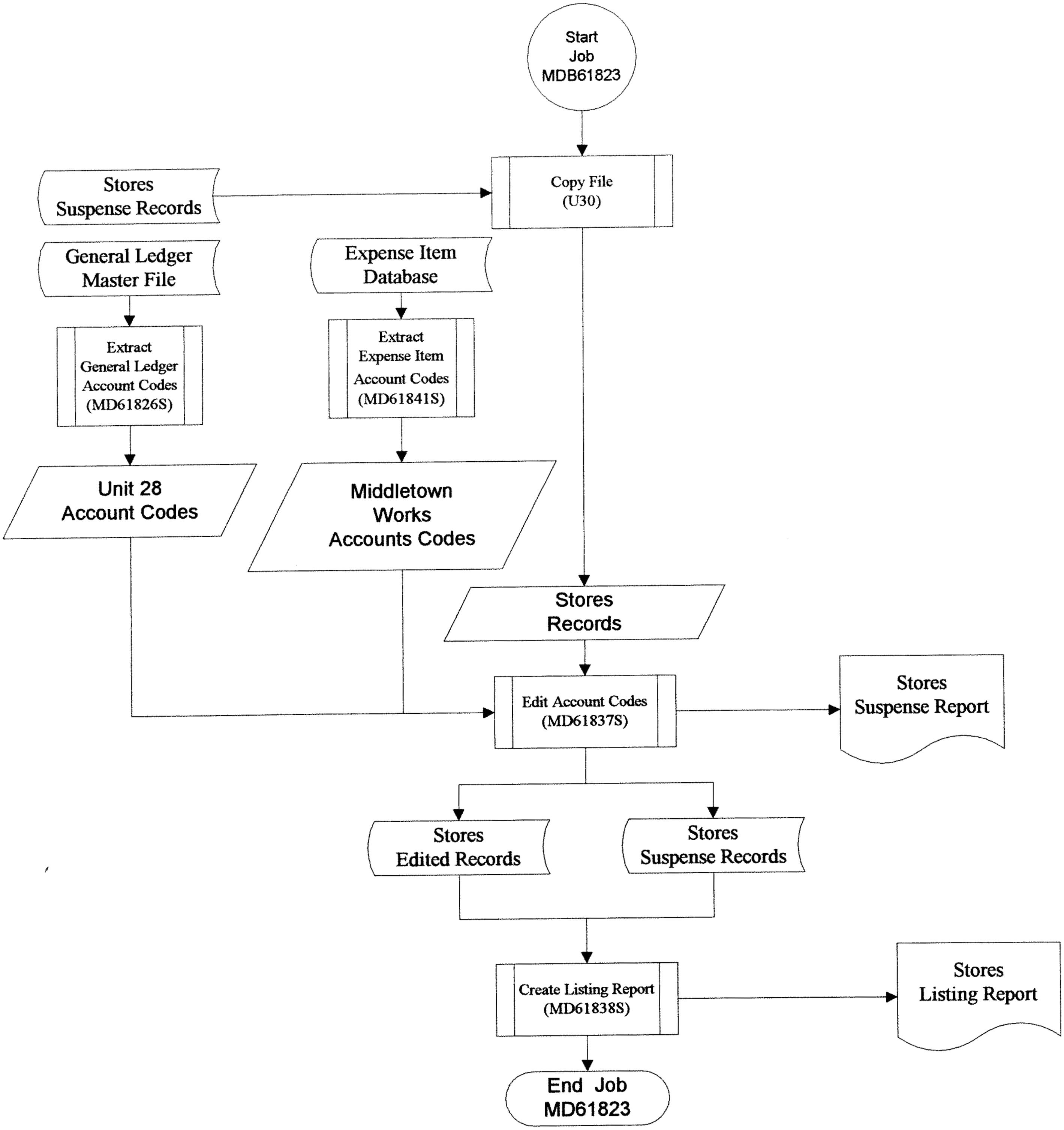
Middletown Works Feeders System

Retrieve Stores Job(MDB61822)


A
General Ledger Master File
Expense Item Database
Extract General Ledger Account Codes (MD61826S)
Extract Expense Item Account Codes (MD61841S)
Unit 28 Account Code
Middletown Works Account Codes
Edit Account Codes (MD61837S)
Stores Edited Records
Stores Suspense Records
Create Listing Report (MD61838S)
Stores Listing Report
Erase File (U10)
Raw Stores Data
End Job MD61822

Structured Chart

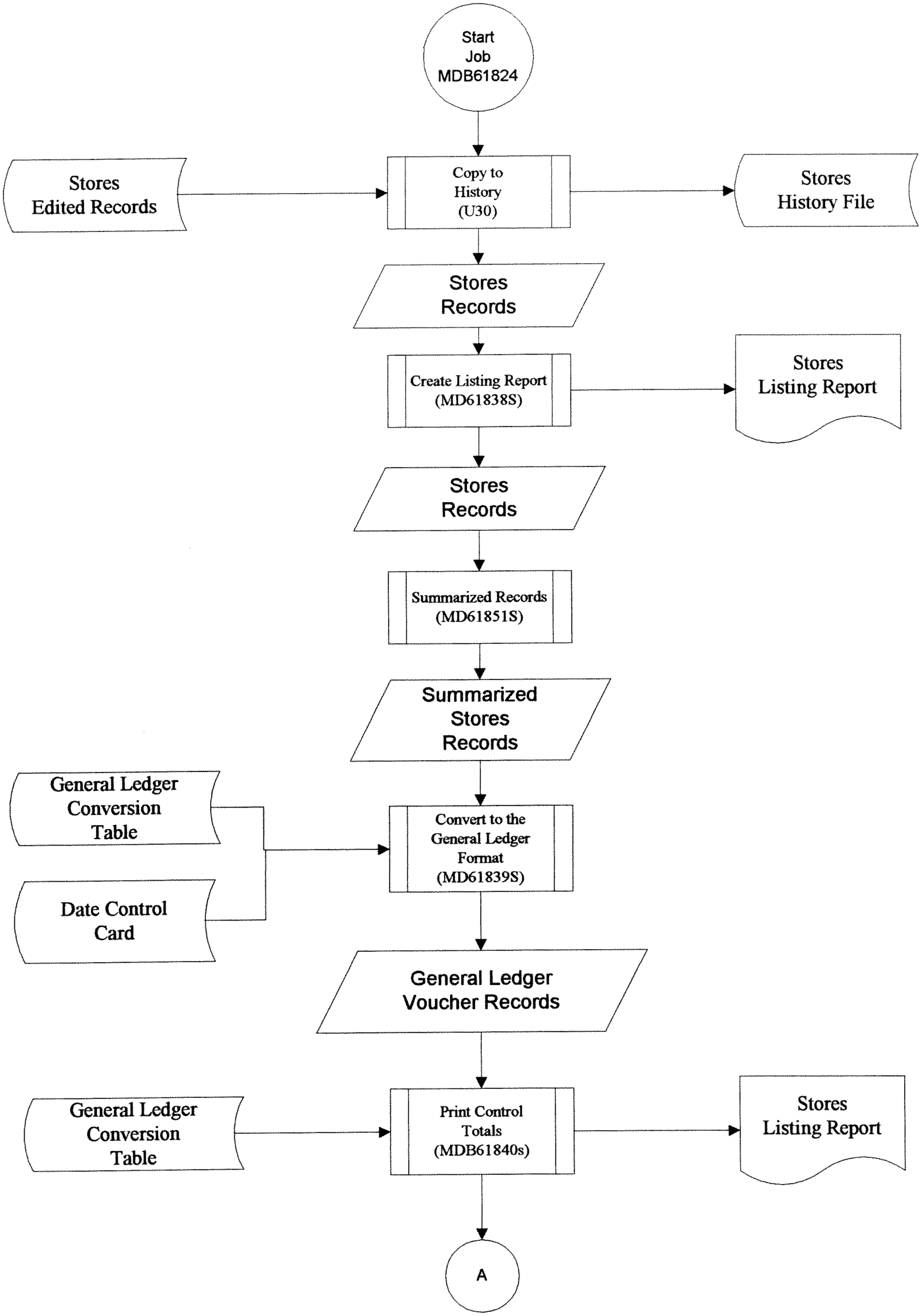
Middletown Works Feeders System
Edit Stores Job (MDB61823)



Process Modeling

Middletown Works Feeders System

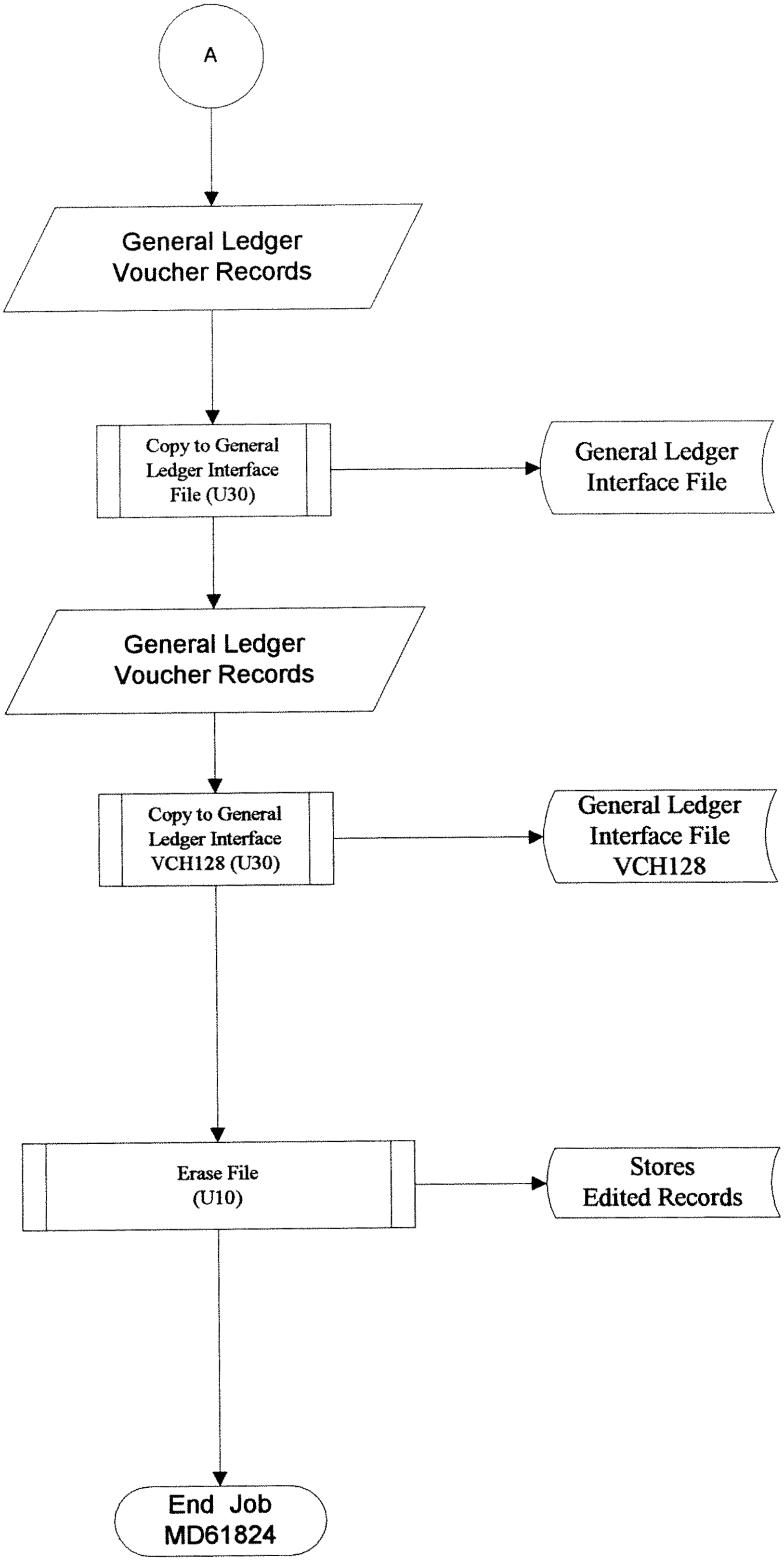
Load Stores Job (MDB61824)


Start Job MDB61824
Stores Edited Records
Copy to History (U30)
Stores History File
Stores Records
Create Listing Report (MD61838S)
Stores Listing Report
Stores Records
Summarized Records (MD61851S)
Summarized Stores Records
General Ledger Conversion Table
Date Control Card
Convert to the General Ledger Format (MD61839S)
General Ledger Voucher Records
General Ledger Conversion Table
Print Control Totals (MDB61840s)
Stores Listing Report
A

Process Modeling

Middletown Works Feeders System

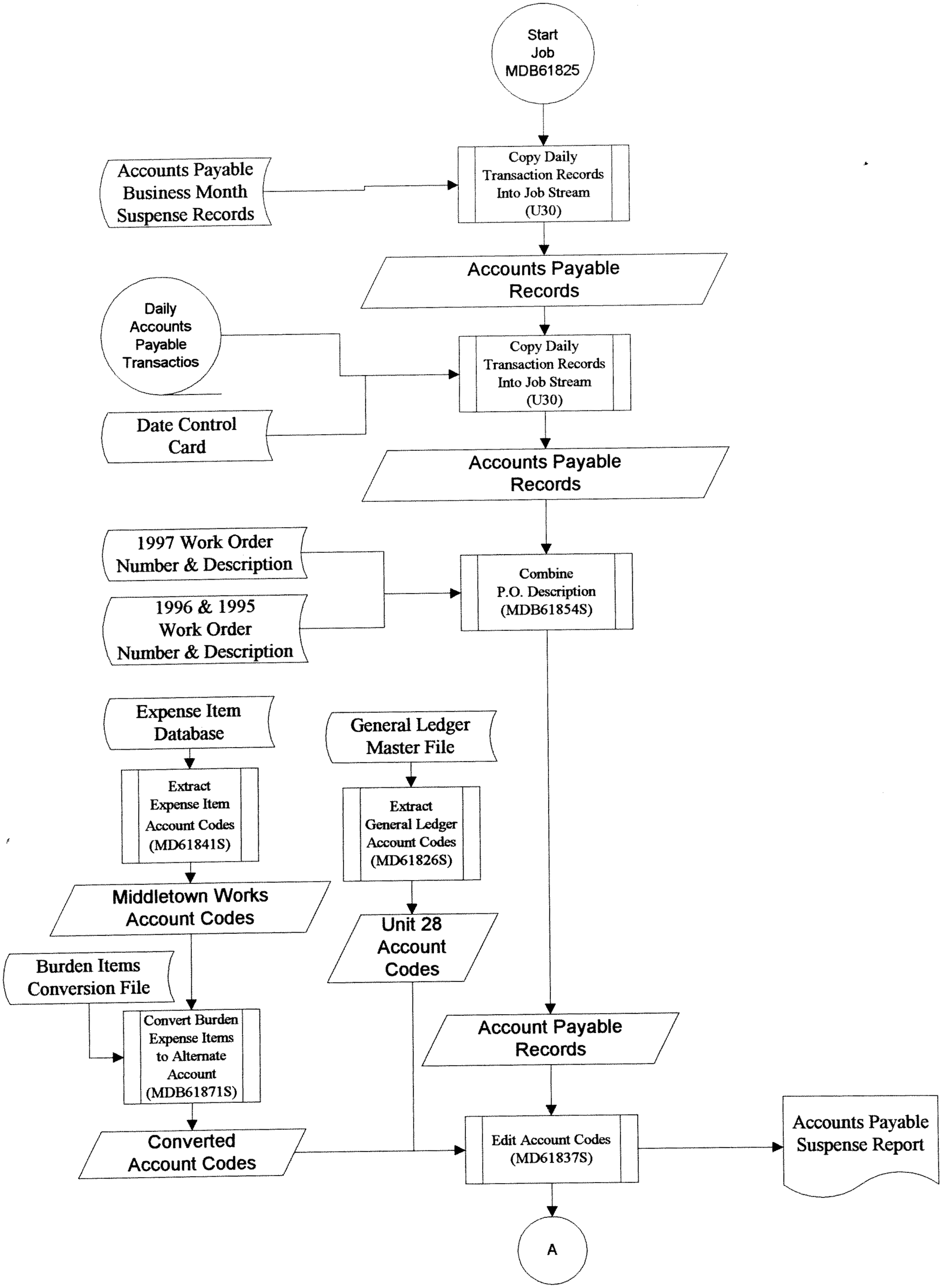
Load Stores Job (MDB61824)


A
General Ledger Voucher Records
Copy to General Ledger Interface File (U30)
General Ledger Interface File
General Ledger Voucher Records
Copy to General Ledger Interface VCH128 (U30)
General Ledger Interface File VCH128
Erase File (U10)
Stores Edited Records
End Job MD61824

Structured Chart

Middletown Works Feeders System

Retrieve Accounts Payable Job(MDB61825)

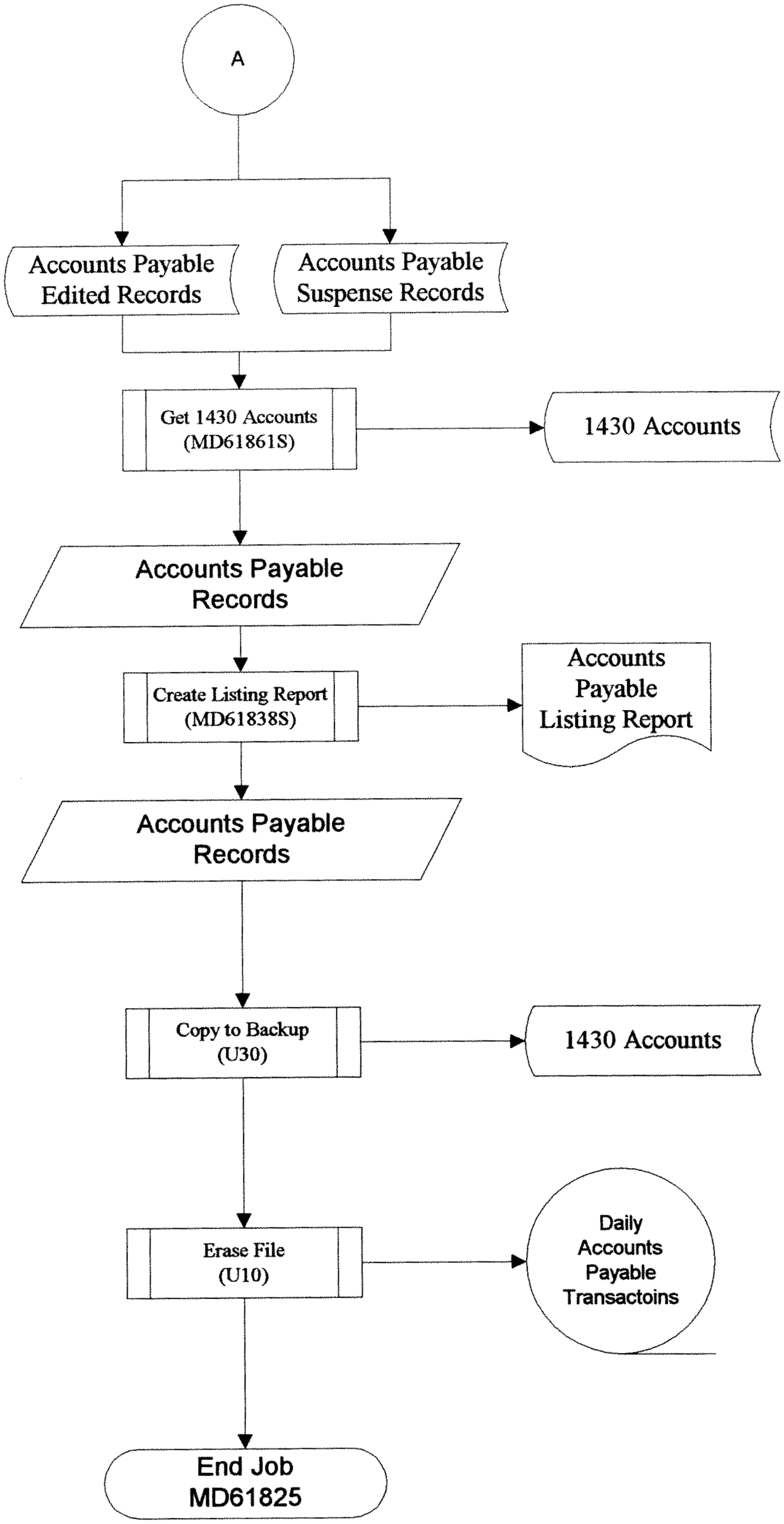


Start Job MDB61825
Accounts Payable Business Month Suspense Records
Copy Daily Transaction Records Into Job Stream (U30)
Accounts Payable Records
Daily Accounts Payable Transactios
Copy Daily Transaction Records Into Job Stream (U30)
Date Control Card
Accounts Payable Records
1997 Work Order Number & Description
1996 & 1995 Work Order Number & Description
Combine P.O. Description (MDB61854S)
Expense Item Database
Extract Expense Item Account Codes (MD61841S)
General Ledger Master File
Extract General Ledger Account Codes (MD61826S)
Middletown Works Account Codes
Unit 28 Account Codes
Burden Items Conversion File
Convert Burden Expense Items to Alternate Account (MDB61871S)
Account Payable Records
Converted Account Codes
Edit Account Codes (MD61837S)
Accounts Payable Suspense Report
A

Structured Chart

Middletown Works Feeders System

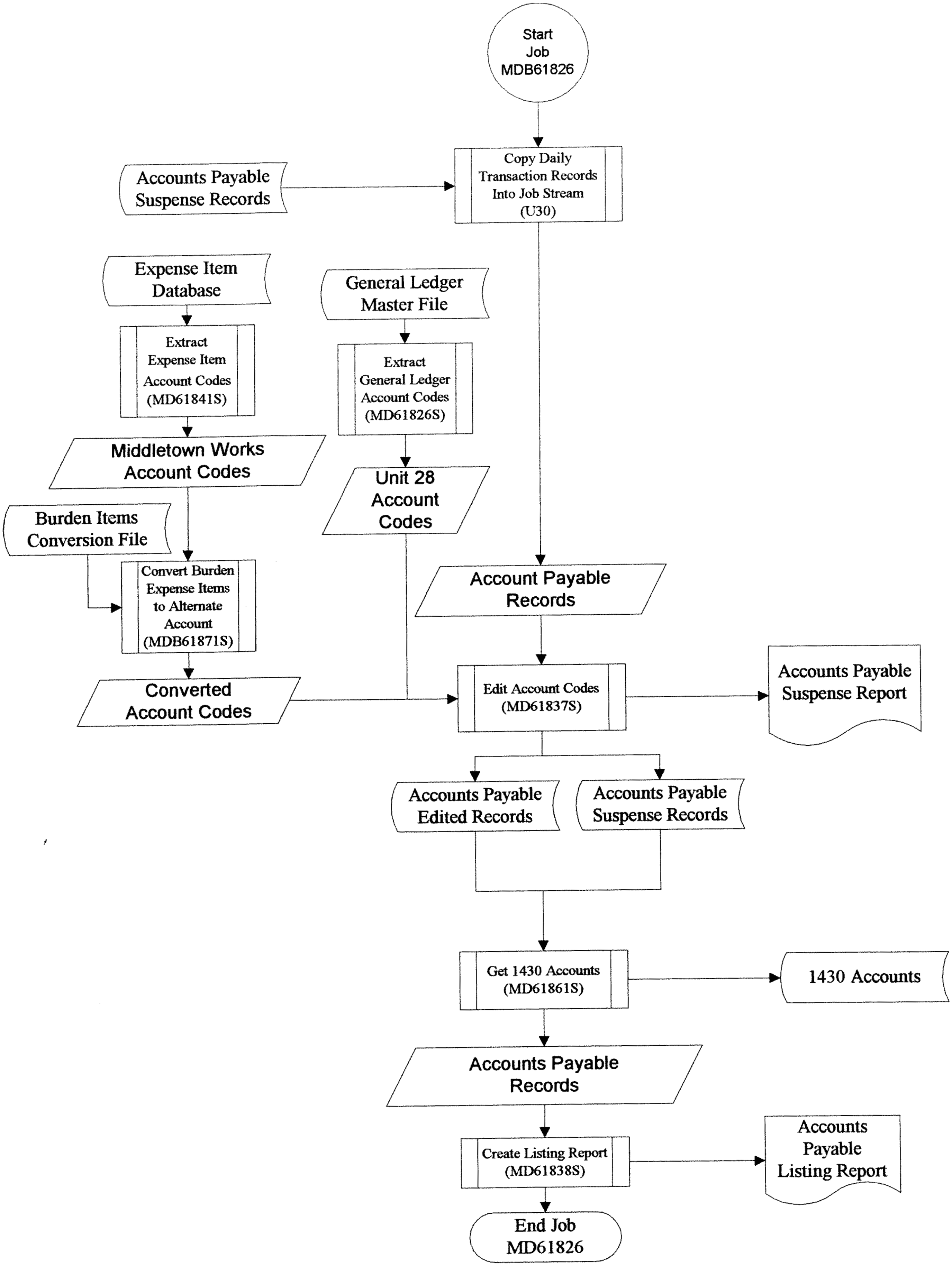
Retrieve Accounts Payable Job(MDB61825)


A
Accounts Payable Edited Records
Accounts Payable Suspense Records
Get 1430 Accounts (MD61861S)
1430 Accounts
Accounts Payable Records
Create Listing Report (MD61838S)
Accounts Payable Listing Report
Accounts Payable Records
Copy to Backup (U30)
1430 Accounts
Erase File (U10)
Daily Accounts Payable Transactoins
End Job MD61825

Structured Chart

Middletown Works Feeders System

Edit Accounts Payable Job(MDB61826)

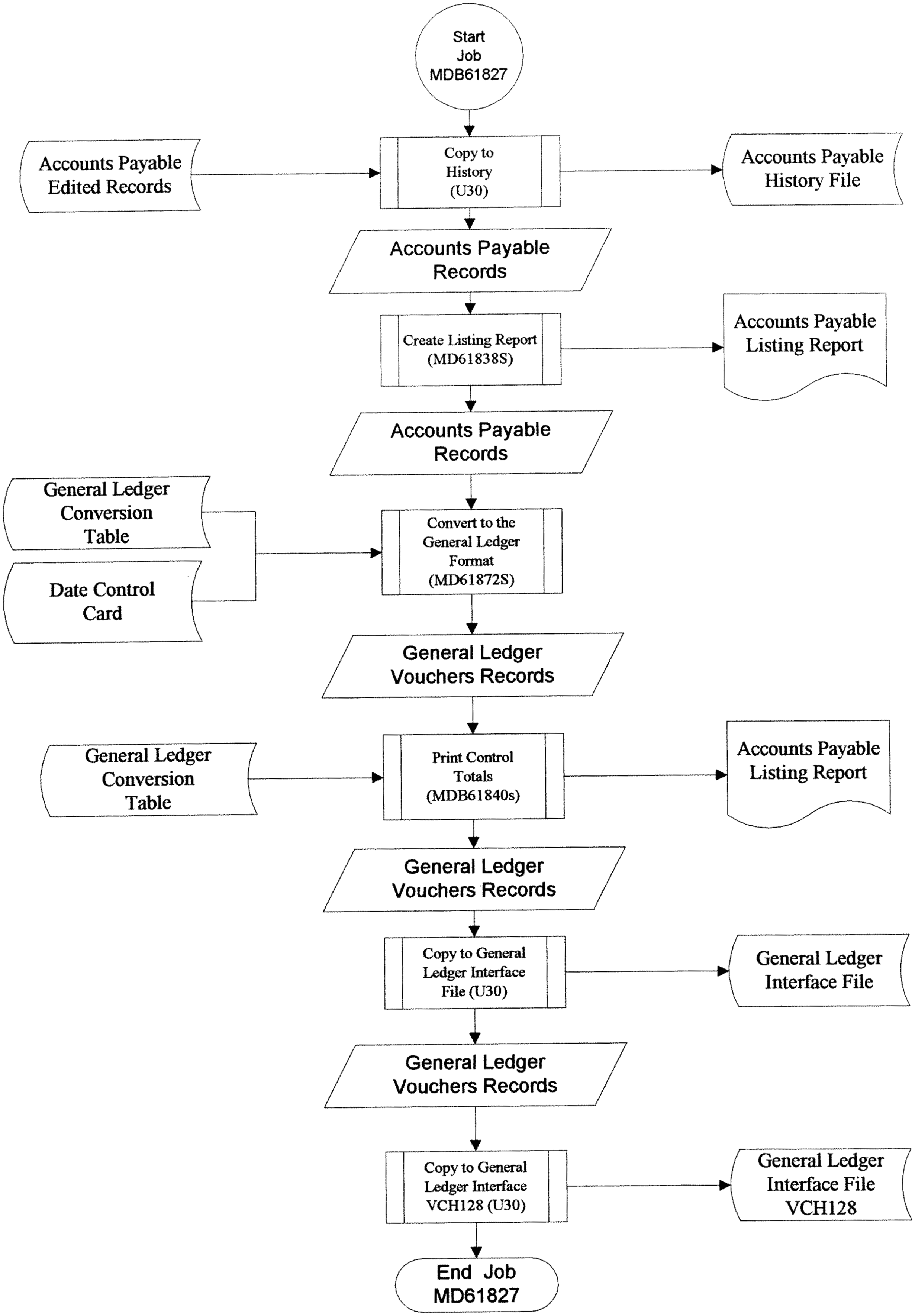


Start Job MDB61826
Accounts Payable Suspense Records
Copy Daily Transaction Records Into Job Stream (U30)
Expense Item Database
General Ledger Master File
Extract Expense Item Account Codes (MD61841S)
Extract General Ledger Account Codes (MD61826S)
Middletown Works Account Codes
Unit 28 Account Codes
Burden Items Conversion File
Convert Burden Expense Items to Alternate Account (MDB61871S)
Account Payable Records
Converted Account Codes
Edit Account Codes (MD61837S)
Accounts Payable Suspense Report
Accounts Payable Edited Records
Accounts Payable Suspense Records
Get 1430 Accounts (MD61861S)
1430 Accounts
Accounts Payable Records
Create Listing Report (MD61838S)
Accounts Payable Listing Report
End Job MD61826

Structured Chart

Middletown Works Feeders System

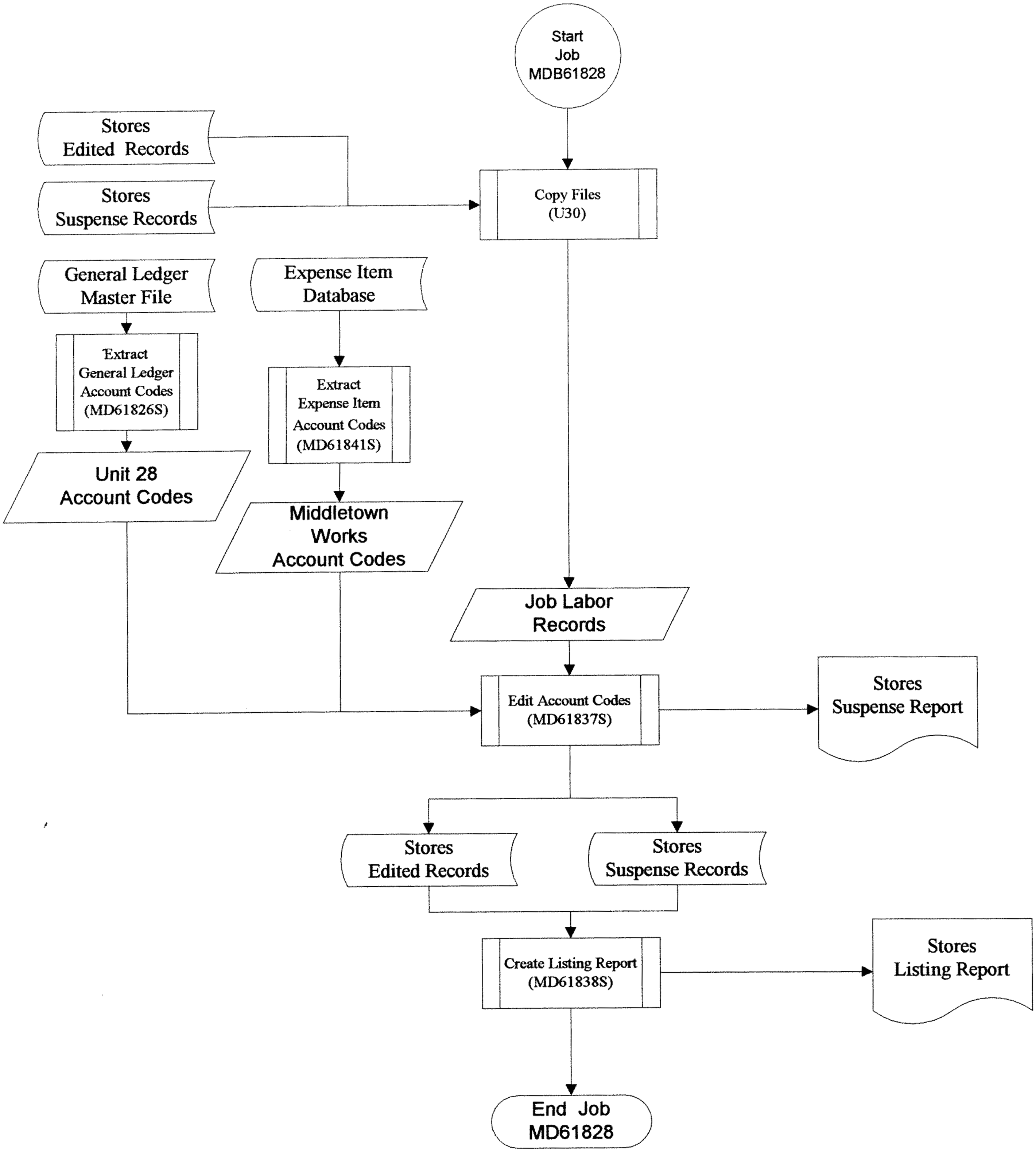
Load Accounts Payable Job (MDB61827)


Start Job MDB61827
Accounts Payable Edited Records
Copy to History (U30)
Accounts Payable History File
Accounts Payable Records
Create Listing Report (MD61838S)
Accounts Payable Listing Report
Accounts Payable Records
General Ledger Conversion Table
Date Control Card
Convert to the General Ledger Format (MD61872S)
General Ledger Vouchers Records
General Ledger Conversion Table
Print Control Totals (MDB61840s)
Accounts Payable Listing Report
General Ledger Vouchers Records
Copy to General Ledger Interface File (U30)
General Ledger Interface File
General Ledger Vouchers Records
Copy to General Ledger Interface VCH128 (U30)
General Ledger Interface File VCH128
End Job MD61827

Structured Chart

Middletown Works Feeders System

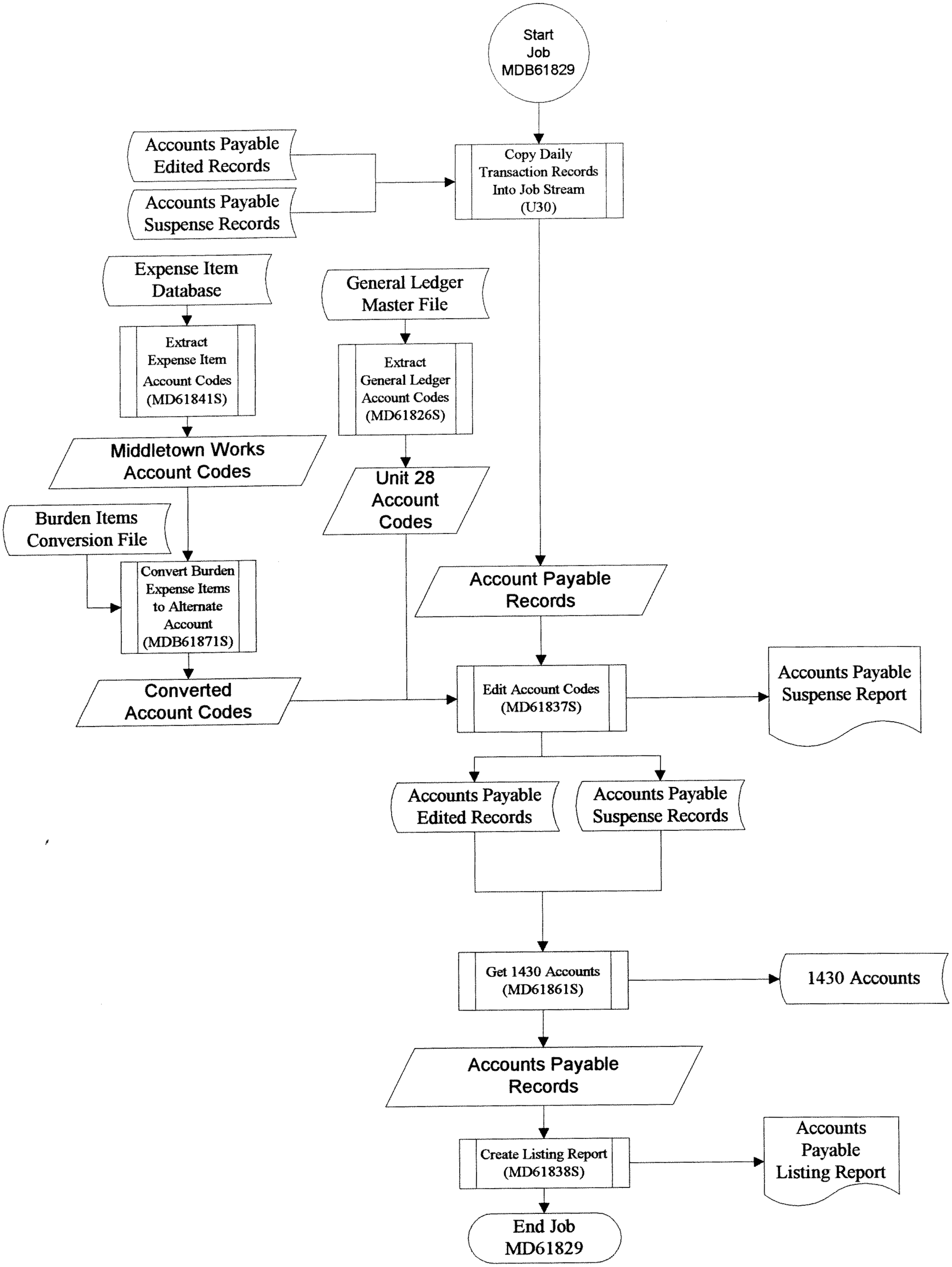
Re-edit Stores Job (MDB61828)



Structured Chart

Middletown Works Feeders System

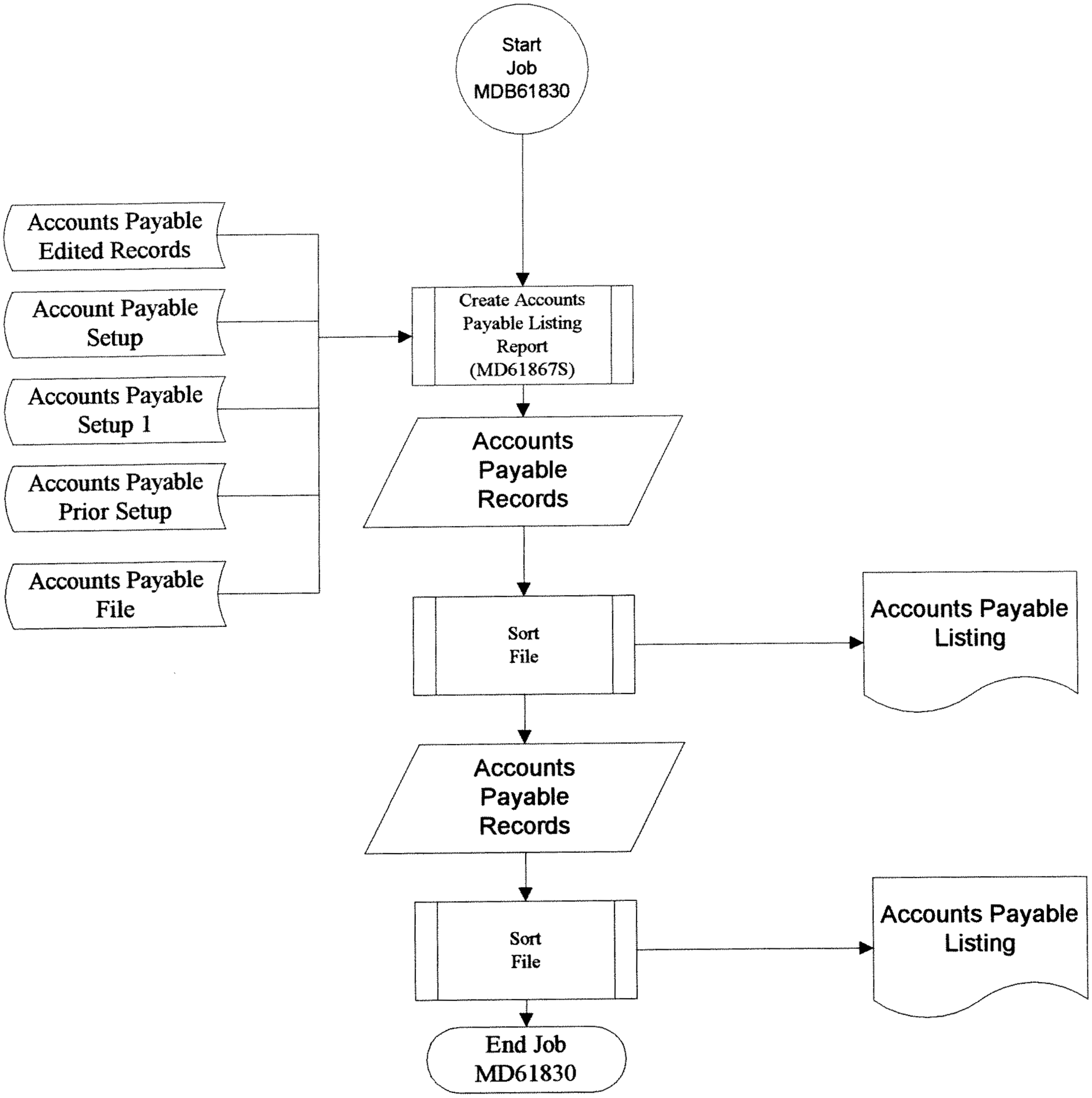
Re-edit Accounts Payable Job(MDB61829)


Start Job MDB61829
Accounts Payable Edited Records
Accounts Payable Suspense Records
Copy Daily Transaction Records Into Job Stream (U30)
Expense Item Database
Extract Expense Item Account Codes (MD61841S)
General Ledger Master File
Extract General Ledger Account Codes (MD61826S)
Middletown Works Account Codes
Unit 28 Account Codes
Burden Items Conversion File
Convert Burden Expense Items to Alternate Account (MDB61871S)
Account Payable Records
Converted Account Codes
Edit Account Codes (MD61837S)
Accounts Payable Suspense Report
Accounts Payable Edited Records
Accounts Payable Suspense Records
Get 1430 Accounts (MD61861S)
1430 Accounts
Accounts Payable Records
Create Listing Report (MD61838S)
Accounts Payable Listing Report
End Job MD61829

Structured Chart

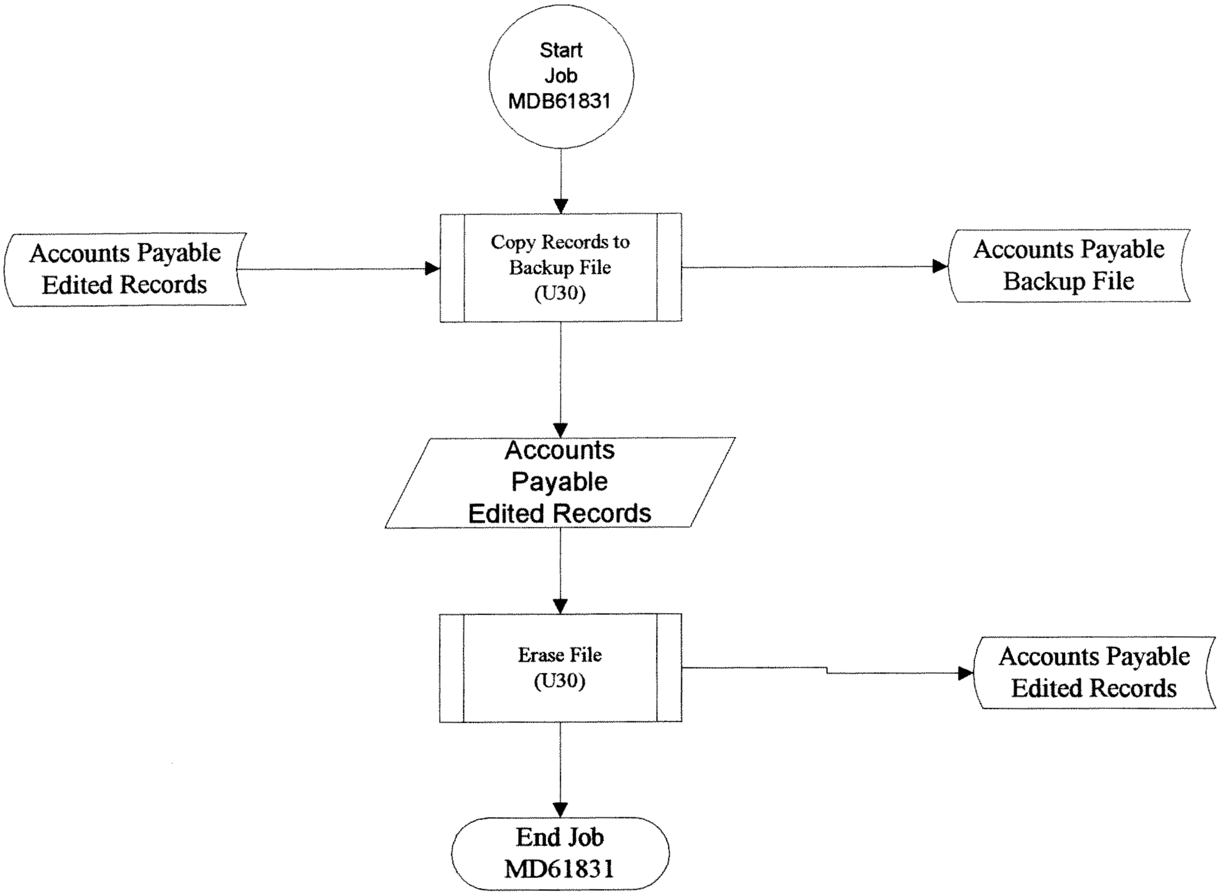
Middletown Works Feeders System

Clears Accounts Payable Job(MDB61830)



Structured Chart

Middletown Works Feeders System
Clears Accounts Payable Job(MDB61831)


Start Job MDB61831
Accounts Payable Edited Records
Copy Records to Backup File (U30)
Accounts Payable Backup File
Accounts Payable Edited Records
Erase File (U30)
Accounts Payable Edited Records
End Job MD61831

Appendix B

Appendix B contains the first page of each program used in constructing the New Feeder system. These program names match the program identification found in the Hierarchy Charts and Structured Flowcharts.

```
MD.CS.ISPF.CLIST(FDRSTART)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:28 MDSCMCM  CPU1          **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSTART)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 0
      /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
      /*  APPLICATION:  MW FEEDERS -- COST SHEETS                   */
      /*   WRITTEN BY:  MICHAEL MENEFIELD, ISSC                     */
      /*         DATE:  MARCH 1996                                  */
      /*      PURPOSE:  THIS FUNCTION IS USED TO:                   */
      /*                (1) INITIALIZE THE APPLICATION'S GLOBAL     */
      /*                    CONSTANT VARIALBES.                     */
      /*                (2) START THE PRIMARY OTIONS MENU.          */
      /*                                                            */
      /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
CONTROL MSG NOFLUSH                    /* CONLIST SYMLIST           */
GLOBAL ZPF01,ZPF02,ZPF03,ZPF04,ZPF05,ZPF06,ZPF07,ZPF08,ZPF09,ZPF10,+
       ZPF11,ZPF12,ZPF13,ZPF14,ZPF15,ZPF16,ZPF17,ZPF18,ZPF19,ZPF20,+
       ZPF21,ZPF22,ZPF23,ZPF24,SPF01,SPF02,SPF03,SPF04,SPF05,SPF06,+
       SPF07,SPF08,SPF09,SPF10,SPF11,SPF12,SPF13,SPF14,SPF15,SPF16,+
       SPF17,SPF18,SPF19,SPF20,SPF21,SPF22,SPF23,SPF24
SET &ZPFCTL = ON
SET &ZPFSET = ALT
SET &ZPFFMT = SIX
ISPEXEC VPUT (ZPFCTL ZPFSET ZPFFMT) PROFILE
SYSCALL PFKYSET
SET LIBR     = MD.CS.ISPF              /* DATASET PREFIX INTERNAL      */
SET SQDS     = MD.SQDSCS               /* DATASET PREFIX EXTERNAL      */

ISPEXEC VPUT (LIBR,SQDS) PROFILE
ISPEXEC SELECT  PANEL(FDR@MENU)
IF &ZCMD ¬= X THEN ISPEXEC CONTROL NONDISPL ENTER
SYSCALL PFKYRES
EXIT

/*******************************************************************/
/* THIS BLOCK RESETS THE USERS PFKEYS                              */
/*******************************************************************/
PFKYSET: +
PROC 0
  ISPEXEC VGET (ZPF01,ZPF02,ZPF03,ZPF04,ZPF05,ZPF06,ZPF07,ZPF08,ZPF09,+
                ZPF10,ZPF11,ZPF12,ZPF13,ZPF14,ZPF15,ZPF16,ZPF17,ZPF18,+
                ZPF19,ZPF20,ZPF21,ZPF22,ZPF23,ZPF24) PROFILE
  SET &SPF01 = &ZPF01
  SET &SPF02 = &ZPF02
  SET &SPF03 = &ZPF03
  SET &SPF04 = &ZPF04
  SET &SPF05 = &ZPF05
  SET &SPF06 = &ZPF06
  SET &SPF07 = &ZPF07
  SET &SPF08 = &ZPF08
  SET &SPF09 = &ZPF09
  SET &SPF10 = &ZPF10
  SET &SPF11 = &ZPF11
  SET &SPF12 = &ZPF12
  SET &SPF13 = &ZPF13
  SET &SPF14 = &ZPF14
  SET &SPF15 = &ZPF15
  SET &SPF16 = &ZPF16
```

```
MD.CS.ISPF.CLIST(FDRADMF)
VPSPRINT 6.1.018   TUESDAY MARCH 4,1997         10:39:09 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRADMF)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 FILEEDIT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE FILE EDIT PROCESS**/
/**FOR CONVERSION FILES.  THE FILE(S) WHICH CAN BE EDITED ON-LINE **/
/**IS (ARE) THE FOLLOWING:                                        **/
/**    - ACCOUNTS PAYABLE CONVERSION TABLE (APCT)                 **/
/**    - ACCOUNTS PAYABLE SPECIAL CONVERSION LIST (APSC)          **/
/**    - HOURLY PAYROLL CONVERSION TABLE (HPCT)                   **/
/**    - SALARY PAYROLL CONVERSION TABLE (SPCT)                   **/
/**    - STORES CONVERSION TABLE (STCT)                           **/
/**    - JOB LABOR CONVERSION TABLE (JLCT)                        **/
/**    - GENERAL LEDGER CONVERSION TABLE (GLCT)                   **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   02/25/97                                               **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON

SYSCALL KEYSET
SELECT(&FILEEDIT)
  WHEN(GLCT) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB61815)')
    ENDO
  WHEN(APCT) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB61825)')
    ENDO
  WHEN(APSC) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB618SP)')
    ENDO
  WHEN(HPCT) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB61813)')
    ENDO
  WHEN(SPCT) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB61816)')
    ENDO
  WHEN(STCT) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB61822)')
    ENDO
  WHEN(JLCT) +
    DO
      ISPEXEC EDIT DATASET('MD.CS.ISPF.INPUT(MDB61819)')
    ENDO
ENDO

SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL PROFILE
SYSCALL KEYRESET
```

```
MD.CS.ISPF.CLIST(FDRDATE)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:24:52 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRDATE)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 JOBNAME
/********************************************************************/
/********************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE DATE FUNCTION    **/
/**WHICH PROCESSES THE DATE CONTROL CARD FOR THE FOLLOWING        **/
/**PROCESSES:                                                     **/
/**  - ACCOUNTS PAYABLE PROCESS                                   **/
/**  - HOURLY PAYROLL PROCESS                                     **/
/**  - SALARY PAYROLL PROCESS                                     **/
/**  - STORES PROCESS                                             **/
/**  - JOB LABOR PROCESS                                          **/
/********************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/********************************************************************/
CONTROL END(ENDO) MSG NOFLUSH        /* CONLIST SYMLIST
SET MSG      =                       /* INITIALIZE MESSAGE PARAMETER */
                                     /* GET THE CONTROL CARD RECORD  */
SET MOLIST1 = &STR(JAN FEB MAR APR MAY JUN)
SET MOLIST2 = &STR(JUL AUG SEP OCT NOV DEC)
ALLOC FILE(CNTLCARD) DA('MD.CS.ISPF.CNTLCARD(&JOBNAME)') SHR REUSE
OPENFILE   CNTLCARD INPUT
GETFILE    CNTLCARD
                                     /* SET RECORD VARIABLES         */
SET MONTH1   = &SUBSTR(1:3,&CNTLCARD)
CLOSFILE   CNTLCARD
SET FINISH   = NO
DO WHILE (&FINISH = NO)
                                     /* BUSINESS MONTH WARNING       */
 ISPEXEC  DISPLAY  PANEL(FDR$DATE)   /* DISPLAY PANEL WITH           */+
                   MSG(&MSG)         /*  MESSAGE PARAMETER           */
 IF &LASTCC   = 8 THEN               /* 8  - END OR RETURN CMD USED  */+
  DO
   SET ZCMD     =                    /*  NULL                        */
   ISPEXEC VGET ZVERB    SHARED
   IF &ZVERB  = END  THEN            /* PF3 OR END          CMD USED  */+
        SET FINISH   = YES           /* SET LOOP EXIT                */
   ELSE
  ENDO
 ELSE
 IF &ZCMD   = UPDATE THEN            /* PF2 OR UPDATE       CMD USED  */+
  DO
   SET ZCMD     =                    /*  NULL                        */
   SET MSG      = CSAC000            /*  UPDATED       MESSAGE       */
   OPENFILE   CNTLCARD OUTPUT
                                     /* SET OUTPUT RECORD            */
   SET CNTLCARD = &STR(&MONTH1)
   PUTFILE    CNTLCARD               /* WRITE THE RECORD             */
   CLOSFILE   CNTLCARD
  ENDO
ENDO                                 /* DO WHILE (&FINISH = NO)      */
FREE  FILE(CNTLCARD)
                                     /*                              */
GOBACK:  EXIT                        /* FUNCTION END                 */
```

```
MD.CS.ISPF.CLIST(FDRHOURA)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:25:31 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRHOURA)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 JOBNAME
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE NAVIGATION FOR   **/
/**THE HOURLY PAYROLL PROCESSES.  THE HOURLY PAYROLL PROCESSES    **/
/**ARE THE FOLLOWING:                                             **/
/**   - RETRIEVE HOURLY PAYROLL DATA                              **/
/**   - EDIT HOURLY PAYROLL INFORMATION                           **/
/**   - LOAD HOURLY PAYROLL INFORMATION                           **/
/**ALSO, THIS PROGRAM WILL MONITOR THE AFOREMENTIONED PROCESSES   **/
/**WHICH ARE SUBMITTED IN A BACKGROUND PROCESS.                   **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*******************************************************************/
SELECT(&JOBNAME)
  /* RETRIEVE HOURLY PAYROLL */
  WHEN(MDB61813) +
    DO
      %FDRDATE &JOBNAME
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* EDIT SUSPENSE HOURLY PAYROLL */
  WHEN(MDB61814) +
    DO
      %FDRHOURF SUSP
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* LOAD HOURLY PAYROLL */
  WHEN(MDB61815) +
    DO
      ISPEXEC DISPLAY PANEL(FDR$SUBT)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
END
```

```
MD.CS.ISPF.CLIST(FDRHOURF)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:01 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRHOURF)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 FILEEDIT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE FILE EDIT PROCESS**/
/**FOR HOURLY PAYROLL FILES.  THE FILE(S) WHICH CAN BE EDITED ON- **/
/**LINE IS (ARE) THE FOLLOWING:                                  **/
/**   - HOURLY PAYROLL SUSPENSE FILE  (SUSP)                     **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                     **/
/**DATE:   11/15/96                                              **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON

SYSCALL KEYSET
SELECT(&FILEEDIT)
  WHEN(SUSP) +
    DO
      ISPEXEC EDIT DATASET('MD.SQDSCS.HOURLY.SUSPENSE')
    ENDO
ENDO

SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL PROFILE
SYSCALL KEYRESET
EXIT

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPF07, +
                 ZPF08,ZPF10,ZPF11) PROFILE
   SET SAV07 =  &STR.(&ZPF07.)
   SET SAV08 =  &STR.(&ZPF08.)
   SET SAV10 =  &STR.(&ZPF10.)
   SET SAV11 =  &STR.(&ZPF11.)
   SET ZPF07 = UP
   SET ZPF08 = DOWN
   SET ZPF10 = LEFT
   SET ZPF11 = RIGHT
   ISPEXEC VPUT (SAV07,SAV08,SAV10,SAV11, +
                 ZPF07,ZPF08,ZPF10,ZPF11) PROFILE
SETEND: +
ENDO

/*********************************************************************/
/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYRESET: +
PROC 0
  ISPEXEC VGET (SAV07,SAV08,SAV10,SAV11) PROFILE
  SET ZPF07  = &STR.(&SAV07.)
```

```
MD.CS.ISPF.CLIST(FDRHOURR)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:10 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRHOURR)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 RPRT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE REPORT DISPLAY   **/
/**AND PRINT FUNCTION FOR THE HOURLY PAYROLL PROCESS. THE         **/
/**FOLLOWING REPORTS ARE DISPLAYED ON-LINE AND CAN BE PRINTED ON  **/
/**REQUEST:                                                       **/
/**   - HOURLY PAYROLL SUSPENSE REPORT (SUSP)                     **/
/**   - HOURLY PAYROLL LISTING REPORT  (LIST)                     **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON
SYSCALL KEYSET

ISPEXEC CONTROL ERRORS RETURN
SELECT(&RPRT)
  WHEN(SUSP) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.HOURLY.SUSPENSE.RPT')
    ENDO
  WHEN(LIST) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.HOURLY.RPT')
    ENDO
ENDO

SET BROWSECC = &LASTCC
ISPEXEC CONTROL ERRORS CANCEL

IF &BROWSECC ¬= 0 THEN DO
   SET ZEDSMSG = &STR(EMPTY FILE)
   SET ZEDLMSG = &STR(NO RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
   GOTO ERRORRTN
ENDO                                 /* DISPLAY PF KEY LABELS  OFF   */

IF &BROWSECC = 0 THEN DO
   SET ZEDSMSG = &STR(FILE BROWSED)
   SET ZEDLMSG = &STR(RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
ENDO
                                     /* DISPLAY PF KEY LABELS  OFF   */
SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL   PROFILE
SYSCALL KEYRESET

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPFL04,ZPF04,ZPF07, +
MD.CS.ISPF.CLIST(FDRSLRYF)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:11 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSLRYF)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 FILEEDIT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE FILE EDIT PROCESS**/
/**FOR SALARY PAYROLL FILES.  THE FILE(S) WHICH CAN BE EDITED     **/
/**ON-LINE IS (ARE) THE FOLLOWING:                                **/
/**   - SALARY PAYROLL SUSPENSE FILE (SUSP)                       **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:    11/15/96                                              **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON

SYSCALL KEYSET
SELECT(&FILEEDIT)
  WHEN(SUSP) +
    DO
      ISPEXEC EDIT DATASET('MD.SQDSCS.SALARY.SUSPENSE')
    ENDO
ENDO

SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL PROFILE
SYSCALL KEYRESET
EXIT

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPF07, +
                ZPF08,ZPF10,ZPF11) PROFILE
   SET SAV07 =  &STR.(&ZPF07.)
   SET SAV08 =  &STR.(&ZPF08.)
   SET SAV10 =  &STR.(&ZPF10.)
   SET SAV11 =  &STR.(&ZPF11.)
   SET ZPF07 = UP
   SET ZPF08 = DOWN
   SET ZPF10 = LEFT
   SET ZPF11 = RIGHT
   ISPEXEC VPUT (SAV07,SAV08,SAV10,SAV11, +
                ZPF07,ZPF08,ZPF10,ZPF11) PROFILE
SETEND: +
ENDO

/*********************************************************************/
/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYRESET: +
PROC 0
  ISPEXEC VGET (SAV07,SAV08,SAV10,SAV11) PROFILE
  SET ZPF07  = &STR.(&SAV07.)
```

```
MD.CS.ISPF.CLIST(FDRSLRYR)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:21 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSLRYR)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 RPRT
/******************************************************************/
/******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE REPORT DISPLAY   **/
/**AND PRINT FUNCTION FOR THE SALARY PAYROLL PROCESS. THE         **/
/**FOLLOWING REPORTS ARE DISPLAYED ON-LINE AND CAN BE PRINTED ON  **/
/**REQUEST:                                                       **/
/**   - SALARY PAYROLL SUSPENSE REPORT (SUSP)                     **/
/**   - SALARY PAYROLL LISTING REPORT  (LIST)                     **/
/******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON
SYSCALL KEYSET

ISPEXEC CONTROL ERRORS RETURN
SELECT(&RPRT)
  WHEN(SUSP) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.SALARY.SUSPENSE.RPT')
    ENDO
  WHEN(LIST) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.SALARY.RPT')
    ENDO
ENDO

SET BROWSECC = &LASTCC
ISPEXEC CONTROL ERRORS CANCEL

IF &BROWSECC ¬= 0 THEN DO
   SET ZEDSMSG = &STR(EMPTY FILE)
   SET ZEDLMSG = &STR(NO RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
   GOTO ERRORRTN
ENDO                                 /* DISPLAY PF KEY LABELS  OFF   */

IF &BROWSECC = 0 THEN DO
   SET ZEDSMSG = &STR(FILE BROWSED)
   SET ZEDLMSG = &STR(RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
ENDO
                                     /* DISPLAY PF KEY LABELS  OFF   */
SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL   PROFILE
SYSCALL KEYRESET

/********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES ********************************/
/********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPFL04,ZPF04,ZPF07, +
MD.CS.ISPF.CLIST(FDRSLRYA)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:04 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSLRYA)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 JOBNAME
/******************************************************************/
/******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE NAVIGATION FOR   **/
/**THE SALARY PAYROLL PROCESSES.  THE SALARY PAYROLL PROCESSES    **/
/**ARE THE FOLLOWING:                                             **/
/**    - RETRIEVE SALARY PAYROLL DATA                             **/
/**    - EDIT SUSPENSE SALARY PAYROLL INFORMATION                 **/
/**    - LOAD SALARY PAYROLL INFORMATION                          **/
/**ALSO, THIS PROGRAM WILL MONITOR THE AFOREMENTIONED PROCESSES   **/
/**WHICH ARE SUBMITTED IN A BACKGROUND PROCESS.                   **/
/******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/******************************************************************/
SELECT(&JOBNAME)
  /* RETRIEVE SALARY */
  WHEN(MDB61816) +
    DO
      %FDRDATE &JOBNAME
      SET STEP = SUBMIT
    END
  /* EDIT SUSPENSE SALARY */
  WHEN(MDB61817) +
    DO
      %FDRSLRYF SUSP
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* LOAD SUSPENSE SALARY */
  WHEN(MDB61818) +
    DO
      ISPEXEC DISPLAY PANEL(FDR$SUBT)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
END

/* SUBMIT THE JOB FROM SKELS    */
SUBMIT 'MD.CS.ISPF.SKELS(&JOBNAME)'

WRITENR PROCESSING REQUEST
SET JOBFLAG = 0
```

```
MD.CS.ISPF.CLIST(FDRJOBA)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:24 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRJOBA)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 JOBNAME
/********************************************************************/
/********************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE NAVIGATION FOR    **/
/**THE JOB LABOR PROCESSES. THE JOB LABOR PROCESSES ARE THE        **/
/**FOLLOWING:                                                      **/
/**   - RETRIEVE JOB LABOR DATA                                    **/
/**   - EDIT SUSPENSE JOB LABOR INFORMATION                        **/
/**   - LOAD JOB LABOR INFORMATION                                 **/
/**ALSO, THIS PROGRAM WILL MONITOR THE AFOREMENTIONED PROCESSES    **/
/**WHICH ARE SUBMITTED IN A BACKGROUND PROCESS.                    **/
/********************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                       **/
/**DATE:   11/15/96                                                **/
/********************************************************************/
SELECT(&JOBNAME)
  /* RETRIEVE JOB LABOR */
  WHEN(MDB61819) +
    DO
      %FDRDATE &JOBNAME
      SET STEP = SUBMIT
    END
  /* EDIT SUSPENSE JOB LABOR */
  WHEN(MDB61820) +
    DO
      %FDRJOBF SUSP
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* LOAD JOB LABOR */
  WHEN(MDB61821) +
    DO
      ISPEXEC DISPLAY PANEL(FDR$SUBT)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
END

/* SUBMIT THE JOB FROM SKELS    */
SUBMIT 'MD.CS.ISPF.SKELS(&JOBNAME)'

SET JOBFLAG = 0
WRITENR PROCESSING REQUEST
DO WHILE &JOBFLAG EQ 0
```

```
MD.CS.ISPF.CLIST(FDRJOBF)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:30 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRJOBF)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 FILEEDIT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE FILE EDIT PROCESS**/
/**FOR JOB LABOR FILES.  THE FILE(S) WHICH CAN BE EDITED ON-LINE  **/
/**IS (ARE) THE FOLLOWING:                                        **/
/**   - JOB LABOR SUSPENSE FILE  (SUSP)                           **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON

SYSCALL KEYSET
SELECT(&FILEEDIT)
  WHEN(SUSP) +
    DO
      ISPEXEC EDIT DATASET('MD.SQDSCS.JOBLABOR.SUSPENSE')
    ENDO
ENDO

SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL PROFILE
SYSCALL KEYRESET
EXIT

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPF07, +
                ZPF08,ZPF10,ZPF11) PROFILE
   SET SAV07 =  &STR.(&ZPF07.)
   SET SAV08 =  &STR.(&ZPF08.)
   SET SAV10 =  &STR.(&ZPF10.)
   SET SAV11 =  &STR.(&ZPF11.)
   SET ZPF07 = UP
   SET ZPF08 = DOWN
   SET ZPF10 = LEFT
   SET ZPF11 = RIGHT
   ISPEXEC VPUT (SAV07,SAV08,SAV10,SAV11, +
                ZPF07,ZPF08,ZPF10,ZPF11) PROFILE
SETEND: +
ENDO

/*********************************************************************/
/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYRESET: +
PROC 0
  ISPEXEC VGET (SAV07,SAV08,SAV10,SAV11) PROFILE
  SET ZPF07   = &STR.(&SAV07.)
```

```
MD.CS.ISPF.CLIST(FDRJOBR)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:36 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRJOBR)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 RPRT
/********************************************************************/
/********************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE REPORT DISPLAY   **/
/**AND PRINT FUNCTION FOR THE JOB LABOR PROCESS. THE              **/
/**FOLLOWING REPORTS ARE DISPLAYED ON-LINE AND CAN BE PRINTED ON  **/
/**REQUEST:                                                       **/
/**    - JOB LABOR SUSPENSE REPORT (SUSP)                         **/
/**    - JOB LABOR LISTING REPORT  (LIST)                         **/
/**    - JOB LABOR MONTHLY REPORT  (MNTH)                         **/
/**    - JOB LABOR CORRECTION REPORT (CURR)                       **/
/********************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/********************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON
SYSCALL KEYSET

ISPEXEC CONTROL ERRORS RETURN
SELECT(&RPRT)
  WHEN(SUSP) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.JOBLABOR.SUSPENSE.RPT')
    ENDO
  WHEN(LIST) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.JOBLABOR.RPT')
    ENDO
  WHEN(MNTH) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.JOBLABOR.MONTHLY.RPT')
    ENDO
  WHEN(SUMM) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.JOBLABOR.SUMMARY.RPT')
    ENDO
ENDO

SET BROWSECC = &LASTCC
ISPEXEC CONTROL ERRORS CANCEL

IF &BROWSECC ¬= 0 THEN DO
   SET ZEDSMSG = &STR(EMPTY FILE)
   SET ZEDLMSG = &STR(NO RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
   GOTO ERRORRTN
ENDO                                  /* DISPLAY PF KEY LABELS  OFF   */

IF &BROWSECC = 0 THEN DO
   SET ZEDSMSG = &STR(FILE BROWSED)
   SET ZEDLMSG = &STR(RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
ENDO
                                      /* DISPLAY PF KEY LABELS  OFF   */
```

```
MD.CS.ISPF.CLIST(FDRSTORA)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:34 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSTORA)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 JOBNAME
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE NAVIGATION FOR   **/
/**THE STORES PROCESSES.  THE STORES PROCESSES ARE THE FOLLOWING: **/
/**   - RETRIEVE STORES DATA                                      **/
/**   - EDIT SUSPENSE STORES INFORMATION                          **/
/**   - LOAD STORES INFORMATION                                   **/
/**ALSO, THIS PROGRAM WILL MONITOR THE AFOREMENTIONED PROCESSES   **/
/**WHICH ARE SUBMITTED IN A BACKGROUND PROCESS.                   **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*******************************************************************/
SELECT(&JOBNAME)
  /* RETRIEVE STORES */
  WHEN(MDB61822) +
    DO
      %FDRDATE &JOBNAME
      SET STEP = SUBMIT
    END
  /* EDIT SUSPENSE STORES */
  WHEN(MDB61823) +
    DO
      %FDRSTORF SUSP
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* LOAD STORES */
  WHEN(MDB61824) +
    DO
      ISPEXEC DISPLAY PANEL(FDR$SUBT)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  WHEN(MDB61828) +
    DO
      %FDRSTORF EDIT
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
```

```
MD.CS.ISPF.CLIST(FDRSTORF)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:41 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSTORF)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 FILEEDIT
/********************************************************************/
/********************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE FILE EDIT PROCESS**/
/**FOR STORES FILES.  THE FILE(S) WHICH CAN BE EDITED ON-LINE     **/
/**IS (ARE) THE FOLLOWING:                                        **/
/**    - STORES SUSPENSE FILE  (SUSP)                             **/
/**    - STORES EDITED FILE    (EDIT)                             **/
/********************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:    11/15/96                                              **/
/********************************************************************/
CONTROL END(ENDO)
SET ZPFCTL    = ON

SYSCALL KEYSET
SELECT(&FILEEDIT)
  WHEN(SUSP) +
    DO
      ISPEXEC EDIT DATASET('MD.SQDSCS.STORES.SUSPENSE')
    ENDO
  WHEN(EDIT) +
    DO
      ISPEXEC EDIT DATASET('MD.SQDSCS.STORES')
    ENDO
ENDO

SET ZPFCTL    = USER
ISPEXEC VPUT ZPFCTL PROFILE
SYSCALL KEYRESET
EXIT

/**********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES **********************************/
/**********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPF07, +
                 ZPF08,ZPF10,ZPF11) PROFILE
   SET SAV07 =  &STR.(&ZPF07.)
   SET SAV08 =  &STR.(&ZPF08.)
   SET SAV10 =  &STR.(&ZPF10.)
   SET SAV11 =  &STR.(&ZPF11.)
   SET ZPF07 = UP
   SET ZPF08 = DOWN
   SET ZPF10 = LEFT
   SET ZPF11 = RIGHT
   ISPEXEC VPUT (SAV07,SAV08,SAV10,SAV11, +
                 ZPF07,ZPF08,ZPF10,ZPF11) PROFILE
SETEND: +
ENDO

/**********************************************************************/
/**********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES **********************************/
```

```
MD.CS.ISPF.CLIST(FDRSTORR)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:27:46 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRSTORR)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

PROC 1 RPRT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE REPORT DISPLAY   **/
/**AND PRINT FUNCTION FOR THE STORES PROCESS. THE FOLLOWING       **/
/**REPORTS ARE DISPLAYED ON-LINE AND CAN BE PRINTED ON USERS      **/
/**REQUEST:                                                       **/
/**   - STORES SUSPENSE REPORT (SUSP)                             **/
/**   - STORES LISTING REPORT  (LIST)                             **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:    11/15/96                                              **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON
SYSCALL KEYSET

ISPEXEC CONTROL ERRORS RETURN
SELECT(&RPRT)
  WHEN(SUSP) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.STORES.SUSPENSE.RPT')
    ENDO
  WHEN(LIST) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.STORES.RPT')
    ENDO
ENDO

SET BROWSECC = &LASTCC
ISPEXEC CONTROL ERRORS CANCEL

IF &BROWSECC ¬= 0 THEN DO
   SET ZEDSMSG = &STR(EMPTY FILE)
   SET ZEDLMSG = &STR(NO RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
   GOTO ERRORRTN
ENDO                                 /* DISPLAY PF KEY LABELS  OFF   */

IF &BROWSECC = 0 THEN DO
   SET ZEDSMSG = &STR(FILE BROWSED)
   SET ZEDLMSG = &STR(RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
ENDO
                                     /* DISPLAY PF KEY LABELS  OFF   */
SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL   PROFILE
SYSCALL KEYRESET

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES **********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPFL04,ZPF04,ZPF07, +
MD.CS.ISPF.CLIST(FDRPAYA)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:42 MDSCMCM  CPU1        **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRPAYA)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 JOBNAME
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE NAVIGATION FOR   **/
/**THE ACCOUNTS PAYABLE PROCESSES. THE ACCOUNTS PAYABLE PROCESSES **/
/**ARE THE FOLLOWING:                                             **/
/**   - RETRIEVE ACCOUNTS PAYABLE DATA                            **/
/**   - EDIT SUSPENSE ACCOUNTS PAYABLE INFORMATION                **/
/**   - LOAD ACCOUNTS PAYABLE INFORMATION                         **/
/**ALSO, THIS PROGRAM WILL MONITOR THE AFOREMENTIONED PROCESSES   **/
/**WHICH ARE SUBMITTED IN A BACKGROUND PROCESS.                   **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*******************************************************************/
SELECT(&JOBNAME)
  /* RETRIEVE ACCOUNTS PAYABLE */
  WHEN(MDB61825) +
    DO
      %FDRDATE &JOBNAME
      SET STEP = SUBMIT
    END
  /* EDIT SUSPENSE ACCOUNTS PAYABLE */
  WHEN(MDB61826) +
    DO
      %FDRPAYF SUSP
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* LOAD ACCOUNTS PAYABLE */
  WHEN(MDB61827) +
    DO
      ISPEXEC DISPLAY PANEL(FDR$SUBT)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
      DO
        ISPEXEC SETMSG MSG(CSAC002)
        GOTO GOBACK
      END
    END
  /* RE-EDIT ACCOUNTS PAYABLE */
  WHEN(MDB61829) +
    DO
      %FDRPAYF EDIT
      ISPEXEC DISPLAY PANEL(CSADSUBJ)
      IF &ZCMD = YES THEN +
         SET STEP = SUBMIT
      ELSE +
MD.CS.ISPF.CLIST(FDRPAYF)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:52 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRPAYF)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 FILEEDIT
/*******************************************************************/
/*******************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE FILE EDIT PROCESS**/
/**FOR JOB LABOR FILES.  THE FILE(S) WHICH CAN BE EDITED ON-LINE  **/
/**IS (ARE) THE FOLLOWING:                                        **/
/**   - ACCOUNTS PAYABLE SUSPENSE FILE (SUSP)                     **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*******************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON

SYSCALL KEYSET
SELECT(&FILEEDIT)
  WHEN(SUSP) +
    DO
      ISPEXEC EDIT DATASET('MD.SQDSCS.ACCTPAY.SUSPENSE')
    ENDO
  WHEN(EDIT) +
    DO
      ISPEXEC EDIT DATASET('MD.AC.ACCPAY')
    ENDO
ENDO

SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL PROFILE
SYSCALL KEYRESET
EXIT

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPF07, +
                 ZPF08,ZPF10,ZPF11) PROFILE
   SET SAV07 =  &STR.(&ZPF07.)
   SET SAV08 =  &STR.(&ZPF08.)
   SET SAV10 =  &STR.(&ZPF10.)
   SET SAV11 =  &STR.(&ZPF11.)
   SET ZPF07 = UP
   SET ZPF08 = DOWN
   SET ZPF10 = LEFT
   SET ZPF11 = RIGHT
   ISPEXEC VPUT (SAV07,SAV08,SAV10,SAV11, +
                 ZPF07,ZPF08,ZPF10,ZPF11) PROFILE
SETEND: +
ENDO

/*********************************************************************/
/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES *********************************/
/*********************************************************************/
```

```
MD.CS.ISPF.CLIST(FDRPAYR)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 7,1997      17:26:58 MDSCMCM  CPU1       **
VPSPRINT 'MD.CS.ISPF.CLIST(FDRPAYR)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
 PAGELEN(60)

PROC 1 RPRT
/*********************************************************************/
/*********************************************************************/
/**THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE REPORT DISPLAY   **/
/**AND PRINT FUNCTION FOR THE ACCOUNTS PAYABLE PROCESS. THE       **/
/**FOLLOWING REPORTS ARE DISPLAYED ON-LINE AND CAN BE PRINTED ON  **/
/**REQUEST:                                                       **/
/**    - ACCOUNTS PAYABLE SUSPENSE REPORT (SUSP)                  **/
/**    - ACCOUNTS PAYABLE LISTING REPORT (LIST)                   **/
/*********************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                      **/
/**DATE:   11/15/96                                               **/
/*********************************************************************/
CONTROL END(ENDO)
SET ZPFCTL   = ON
SYSCALL KEYSET

ISPEXEC CONTROL ERRORS RETURN
SELECT(&RPRT)
  WHEN(SUSP) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.ACCTPAY.SUSPENSE.RPT')
    ENDO
  WHEN(LIST) +
    DO
      ISPEXEC BROWSE DATASET('MD.SQDSCS.ACCTPAY.RPT')
    ENDO
ENDO

SET BROWSECC = &LASTCC
ISPEXEC CONTROL ERRORS CANCEL

IF &BROWSECC ¬= 0 THEN DO
   SET ZEDSMSG = &STR(EMPTY FILE)
   SET ZEDLMSG = &STR(NO RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
   GOTO ERRORRTN
ENDO                                 /* DISPLAY PF KEY LABELS  OFF   */

IF &BROWSECC = 0 THEN DO
   SET ZEDSMSG = &STR(FILE BROWSED)
   SET ZEDLMSG = &STR(RECORDS FOUND IN THE FILE)
   ISPEXEC SETMSG MSG(ISRZ001)
ENDO
                                     /* DISPLAY PF KEY LABELS  OFF   */
SET ZPFCTL   = USER
ISPEXEC VPUT ZPFCTL   PROFILE
SYSCALL KEYRESET

/*********************************************************************/
/** SET KEY FUNCTION AND LABEL NAMES ***********************************/
/*********************************************************************/
KEYSET: +
PROC 0
   ISPEXEC VGET (ZPFL04,ZPF04,ZPF07, +
MD.SIG.ACCT.SOURCE(MD61813S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:13:04 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61813S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 /* BATCH */
 M61813S: PROC OPTIONS(MAIN);
 /********************************************************************/
 /********************************************************************/
 /* THE PURPOSE OF THIS PROGRAM IS TO CONVERT THE HOURLY AND SALARY **/
 /* PAYROLL FEEDERS RECORDS TO THE ACCOUNTING FORMAT.               **/
 /* THIS PROGRAM CONVERTS THE INPUT FILE DATA TO THE GENERAL LEDGER **/
 /* FILE FORMAT.  THIS PROGRAM WILL USE A CONTROL FILE TO           **/
 /* PROVIDE THE LOGIC TO CONVERT THE INPUT DATA INTO USABLE         **/
 /* ACCOUNTING CODES, COST CENTER, EXPENSE ITEMS, SUB ACCOUNTS,     **/
 /* PRODUCT CODES AND PROJECT CODES.                                **/
 /********************************************************************/
 /********************************************************************/
 /**AUTHOR: MICHAEL C. MENEFIELD                                     **/
 /**DATE: 11\7\31                                                   **/
 /********************************************************************/

 /********************************************************************/
 /* FILE DECLARATIONS                                               **/
 /********************************************************************/
  DCL FEEDER  FILE RECORD INPUT;
  DCL FEEDERS FILE RECORD OUTPUT;
 /********************************************************************/
 /* MIDDLETOWN WORKS FEEDERS INPUT LAYOUT:                          **/
 /********************************************************************/
  DCL 1 FEEDER_IN,
        2 SYSTEM_CODE       CHAR(1)         INIT(' '),
        2 LEDGER_CODE       CHAR(1)         INIT(' '),
        2 COST_CENTER       CHAR(4)         INIT(' '),
        2 EXPENSE_CODE      CHAR(4)         INIT(' '),
        2 SUB_ACCT          CHAR(4)         INIT(' '),
        2 DOLLARS           FIXED DEC(9)    INIT(0),
        2 HOURS             FIXED DEC(9,1)  INIT(0),
        2 QUANTITY          FIXED DEC(11,2) INIT(0),
        2 UNITS_OF_MEASURE  CHAR(2)         INIT(' '),
        2 COMODITY_CODE     CHAR(2)         INIT(' '),
        2 FILLER            CHAR(7)         INIT(' ');

 /*******************************************************************/
 /* GENERAL LEDGER RECORD LAYOUT                                   **/
 /*******************************************************************/
1DCL
  1 FDR_FILE,
    2 FDR_KEY1,
      3 FDR_SOURCE       CHAR(2)    INIT(' '), /* FEEDER SOURCE    */
      3 FILLER           CHAR(1)    INIT(' '),
      3 FDR_ACCOUNT      CHAR(4)    INIT(' '), /* GEN LEDGER ACCT  */
      3 FILLER1          CHAR(1)    INIT('-'),
      3 FDR_CENTER       CHAR(4)    INIT(' '), /* COST CENTER CODE */
      3 FILLER2          CHAR(1)    INIT('-'),
      3 FDR_EXPCODE      CHAR(4)    INIT(' '), /* EXPENSE CODE     */
      3 FILLER3          CHAR(1)    INIT('-'),
      3 FDR_SUBACCT      CHAR(6)    INIT(' '), /* SUB ACCT CODE    */
      3 FILLER4          CHAR(1)    INIT('-'),
      3 FDR_PROJECT      CHAR(4)    INIT(' '), /* PROJECT NO       */
```

```
MD.SIG.ACCT.SOURCE(MD61820S)
VPSPRINT 6.1.018   THURSDAY FEBRUARY 27,1997   23:26:11 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61820S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS MISSING = ' ' NOCENTER NONOTES NOSOURCE;

/********************************************************************/
/********************************************************************/
/** THIS PROGRAM MERGES THE PAYROLL HEAD COUNT INTO THE SALARY     **/
/** FILE FEEDER.                                                   **/
/********************************************************************/
/********************************************************************/
/**AUTHOR: MICHAEL C. MENEFIELD                                    **/
/**DATE: 4/19/96                                                   **/
/********************************************************************/

/********************************************************************/
/********************************************************************/
/** PAYROLL HEAD COUNT FILE                                        **/

   DATA EMPCNT;
     INFILE EMPOUT;
     INPUT @001 MILL        $CHAR01.
           @002 CST_CNTR    4.
           @006 CST_CNT     ZD8.;

     IF MILL  = 'M';
     SYS_CODE = 'P';
     LEDGER   = 'O';
     EXPENSE  = 0101;

   PROC SORT DATA=EMPCNT;
     BY SYS_CODE LEDGER CST_CNTR EXPENSE;
/**                                                                **/
/********************************************************************/


/********************************************************************/
/********************************************************************/
/** SALARY FILE                                                    **/

   DATA MWSALARY;
     INFILE MWSALY;
     INPUT @001 SYS_CODE    $CHAR01.
           @002 LEDGER      $CHAR01.
           @003 CST_CNTR    4.
           @007 EXPENSE     4.
           @011 SUBCODE     4.
           @015 DOLLARS     PD5.
           @020 HOURS       PD5.
           @025 QUANTITY    PD6.2
           @031 UOM         $CHAR02.
           @033 FILLER      $CHAR09.;

   PROC SORT DATA=MWSALARY;
     BY SYS_CODE LEDGER CST_CNTR EXPENSE;
/**                                                                **/
/********************************************************************/
```

```
MD.SIG.ACCT.SOURCE(MD61826S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997       18:09:28 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61826S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 M61826S: PROC OPTIONS (MAIN) REORDER;
 /**********************************************************************/23879300
 /**********************************************************************/23879300
 /* THIS PROGRAM SEARCHES THE GENERAL LEDGER MASTER FILE               */23879408
 /* AND EXTRACTS THE MASTER KEY FOR UNIT 28.  THIS PROGRAM WILL        */23879408
 /* PRODUCE A FILE WHICH CONTAINS VALID GENERAL LEDGER MASTER KEY      */
 /* RECORDS.  THESE RECORDS CONTAINS THE FOLLOWING FIELD, ACCOUNTING   */
 /* CODES, COST CENTER, EXPENSE CODES, SUB ACCOUNT CODES, PRODUCT      */
 /* CODES AND PROJECT CODES.                                           */
 /**********************************************************************/23879508
 /**********************************************************************/23879508
 /**AUTHOR: MICHAEL C. MENEFIELD                                      **/23879508
 /**DATE:   1/1/97                                                    **/23879508
 /**********************************************************************/23879508

 DCL LEDGER FILE RECORD SEQUENTIAL KEYED ENV(VSAM GENKEY);                23880200
 DCL MASTER FILE RECORD OUTPUT;                                           23880200

 DCL  1 DATABASE_IO,
        2 F1                     CHAR(8),
        2 UNIT                   CHAR(2)  INIT(' '),
        2 MASTERKEY              CHAR(42) INIT(' '),
        2 F2                     CHAR(87),
        2 OPEN                   CHAR(1)  INIT(' '),
        2 F3                     CHAR(646),
        2 AMT                    CHAR(1)  INIT(' '),
        2 F4                     CHAR(5),
        2 STAT                   CHAR(1)  INIT(' '),
        2 F5                     CHAR(3296);

 DCL  1 RECORD_IO,
        2 KEY1                   CHAR(2)  INIT(' '),
        2 KEY2                   CHAR(42) INIT(' '),
        2 FILLER                 CHAR(1)  INIT(' '),
        2 KEY3                   CHAR(1)  INIT(' '),
        2 KEY4                   CHAR(1)  INIT(' ');
  DCL OFF           BIT(1)           INIT('0'B);
  DCL ON            BIT(1)           INIT('1'B);
  DCL EOF_LEDGER    BIT(1)           INIT('0'B);
1/*********************************************************************/  23884108
 /*********************************************************************/  23884108
 /* INITIALIZE : MAIN PROCEDURE                                       */  23884108
 /*********************************************************************/  23884108
 ON ENDFILE(LEDGER) EOF_LEDGER = ON;
 READ FILE(LEDGER) INTO(DATABASE_IO) KEY('VLMPARNT28');

 DO WHILE (¬EOF_LEDGER & UNIT = '28' & F1 = 'VLMPARNT');
   IF OPEN = '0' THEN
      DO;
         KEY1     = UNIT;
         KEY2     = MASTERKEY;
         KEY3     = AMT;
         KEY4     = STAT;
         WRITE FILE (MASTER) FROM (RECORD_IO);
         RECORD_IO = ' ';
```

```
MD.SIG.ACCT.SOURCE(MD61833S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:04:06 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61833S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 /* BATCH */
 M61833S: PROC OPTIONS(MAIN);
 /*********************************************************************/
 /*********************************************************************/
 /* THE PURPOSE OF THIS PROGRAM IS TO ACCEPT THE INPUT OF JOB LABOR **/
 /* AND CONVERTS THE INPUT FILE DATA TO THE ACCOUNTING              **/
 /* HISTORY FILE FORMAT.  THIS PROGRAM WILL USE A CONTROL FILE TO   **/
 /* PROVIDE THE LOGIC TO CONVERT THE INPUT DATA INTO USABLE         **/
 /* ACCOUNTING CODES, COST CENTER, EXPENSE ITEMS, SUB ACCOUNTS,     **/
 /* PRODUCT CODES AND PROJECT CODES.                                **/
 /*********************************************************************/
 /*********************************************************************/
 /**AUTHOR: MICHAEL C. MENEFIELD                                     **/
 /**DATE  : 8\23\96                                                  **/
 /*********************************************************************/
 /*********************************************************************/

 /*********************************************************************/
 /* FILE DECLARATIONS                                               **/
 /*********************************************************************/
  DCL FEEDER  FILE RECORD INPUT;
  DCL FEEDERS FILE RECORD OUTPUT;
 /*********************************************************************/
 /* MIDDLETOWN WORKS FEEDERS INPUT LAYOUT:                          **/
 /*********************************************************************/
  DCL 1 FEEDER_IN,
        2 SYSTEM_CODE      CHAR(1)         INIT(' '),
        2 LEDGER_CODE      CHAR(1)         INIT(' '),
        2 COST_CENTER      CHAR(4)         INIT(' '),
        2 EXPENSE_CODE     CHAR(4)         INIT(' '),
        2 SUB_ACCT         CHAR(4)         INIT(' '),
        2 DOLLARS          FIXED DEC(9,2)  INIT(0),
        2 HOURS            FIXED DEC(9,1)  INIT(0),
        2 RATES            CHAR(5)         INIT(' '),
        2 SHOPONLY_HOURS   CHAR(1)         INIT(' '),
        2 UNITS_OF_MEASURE CHAR(2)         INIT(' '),
        2 COMODITY_CODE    CHAR(2)         INIT(' '),
        2 WORK_NUMBER      CHAR(6)         INIT(' '),
        2 WORK_DESCRIPTION CHAR(40)        INIT(' '),
        2 SHOP             CHAR(3)         INIT(' '),
        2 SHOP_NAME        CHAR(25)        INIT(' '),
        2 BADGE            CHAR(5)         INIT(' '),
        2 SEQUENCE_NO      CHAR(10)        INIT(' '),
        2 FILLER           CHAR(1)         INIT(' ');

 /********************************************************************/
 /* GENERAL LEDGER RECORD LAYOUT                                    **/
 /********************************************************************/
1DCL
   1 FDR_FILE,
    2 FDR_KEY1,
      3 FDR_SOURCE       CHAR(2)    INIT(' '), /* FEEDER SOURCE    */
      3 FILLER           CHAR(1)    INIT(' '),
      3 FDR_ACCOUNT      CHAR(4)    INIT(' '), /* GEN LEDGER ACCT  */
      3 FILLER1          CHAR(1)    INIT('-'),
MD.SIG.ACCT.SOURCE(MD61835S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:44:16 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61835S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 /* BATCH */
 M61835S: PROC OPTIONS(MAIN);
 /********************************************************************/
 /********************************************************************/
 /* THE PURPOSE OF THIS PROGRAM IS TO ACCEPT THE DAILY STORES DATA  **/
 /* AND TO CONVERT THE INPUT FILE DATA TO THE STORES HISTORY FILE   **/
 /* HISTORY FILE FORMAT.  THIS PROGRAM WILL USE A CONTROL FILE TO   **/
 /* PROVIDE THE LOGIC TO CONVERT THE INPUT DATA INTO USABLE         **/
 /* ACCOUNTING CODES, COST CENTER, EXPENSE ITEMS, SUB ACCOUNTS,     **/
 /* THESE RECORDS WILL MAKE-UP STORES, OPERATING AND SPARES         **/
 /* INFORMATION.                                                    **/
 /********************************************************************/
 /********************************************************************/
 /**AUTHOR: MICHAEL C. MENEFIELD                                     **/
 /**DATE:   11/26/96                                                 **/
 /********************************************************************/
 /********************************************************************/

 /********************************************************************/
 /* FILE DECLARATIONS                                               **/
 /********************************************************************/
  DCL FEEDER   FILE RECORD INPUT;
  DCL FEEDERS  FILE RECORD OUTPUT;
 /********************************************************************/
 /* MIDDLETOWN WORKS FEEDERS INPUT LAYOUT:-- STORES                 **/
 /********************************************************************/
  DCL 1 FEEDER_IN,
        2 ACCOUNT_CODE     CHAR(4)         INIT(' '),
        2 LEDGER_CODE      CHAR(1)         INIT(' '),
        2 COST_CENTER      CHAR(4)         INIT(' '),
        2 EXPENSE_CODE     CHAR(4)         INIT(' '),
        2 SUB_ACCT         CHAR(4)         INIT(' '),
        2 TRANSACTION_DATE CHAR(5)         INIT(' '),
        2 FILLER           CHAR(1)         INIT(' '),
        2 TRANSACTION_TIME CHAR(5)         INIT(' '),
        2 REC_NO           CHAR(8)         INIT(' '),
        2 BADGE_NO         CHAR(5)         INIT(' '),
        2 STOCK_NO         CHAR(6)         INIT(' '),
        2 UNITS_OF_MEASURE CHAR(2)         INIT(' '),
        2 FILLER2          CHAR(2)         INIT(' '),
        2 QUANTITY         PIC '(7)9'      INIT(0),
        2 DOLLARS          PIC '(6)S9V99'  INIT(0),
        2 NOUN_DESCRIP     CHAR(15)        INIT(' '),
        2 QUALIFY_DESCRIP  CHAR(15)        INIT(' '),
        2 FILLER3          CHAR(23)        INIT(' ');

  DCL  COMODITY_CODE    CHAR(2)         INIT(' ');
  DCL  SYSTEM_CODE      CHAR(1)         INIT('I');
  DCL  HOURS            PIC '(7)9'      INIT(0);
 /*******************************************************************/
 /* GENERAL LEDGER RECORD LAYOUT                                   **/
 /*******************************************************************/
1DCL
  1 FDR_FILE,
    2 FDR_KEY1,
MD.SIG.ACCT.SOURCE(MD61836S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:41:40 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61836S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 M61836S: PROC OPTIONS (MAIN) REORDER;
 /*********************************************************************/23879300
 /*********************************************************************/23879300
 /* THIS PROGRAM WILL SEPARATE THE DAILY STORES RECORDS BASED ON THE  */23879408
 /* BUSINESS MONTH. THIS ENTIALS BUILDING A FILE WITH THE CURRENT     */23879408
 /* BUSINESS MONTH RECORDS AND A SUSPENSE FILE CONTAINING OTHER       */23879408
 /* BUSINESS MONTH RECORDS.                                           */23879408
 /*********************************************************************/23879508
 /*********************************************************************/23879508
 /* AUTHOR: MICHAEL C. MENEFIELD                                      */23879508
 /* DATE:   7/30/96                                                   */23879508
 /*********************************************************************/23879508
 /*********************************************************************/23879508

 DCL  STORES                           FILE RECORD INPUT;                 23880200
 DCL  MATCH                            FILE RECORD OUTPUT;                23880200
 DCL  NOMATCH                          FILE RECORD OUTPUT;                23880200

 DCL  1 RAW_RECORD_IO,
        2 FILLER1   CHAR(9)   INIT(' '),
        2 EXPENSE   CHAR(4)   INIT(' '),
        2 PROJECT   CHAR(4)   INIT(' '),
        2 RAW_MONTH CHAR(2)   INIT(' '),
        2 FILLER2   CHAR(101) INIT(' ');

  DCL OFF           BIT(1)          INIT('0'B);
  DCL ON            BIT(1)          INIT('1'B);
  DCL EOF_STORES    BIT(1)          INIT('0'B);
  DCL MONTH         CHAR(2)         INIT(' ');

1/********************************************************************/  23884108
 /********************************************************************/  23884108
 /* INITIALIZE : MAIN PROCEDURE                                      */  23884108
 /********************************************************************/  23884108
 ON ENDFILE(STORES) EOF_STORES = ON;
 CALL GET_CONTROL_MONTH;
 DO WHILE (¬EOF_STORES);
   READ FILE (STORES) INTO (RAW_RECORD_IO);
   IF (¬EOF_STORES) THEN
    DO;
      IF (SUBSTR(EXPENSE,1,1) = '-') THEN
        DO;
          EXPENSE = SUBSTR(EXPENSE,2,3) || SUBSTR(PROJECT,1,1);
          PROJECT = '0000';
        END;
       IF RAW_MONTH = MONTH THEN
           WRITE FILE (MATCH)   FROM (RAW_RECORD_IO);
       ELSE WRITE FILE (NOMATCH) FROM (RAW_RECORD_IO);
    END;
 END;

1/********************************************************************/  23884108
 /********************************************************************/  23884108
 /* GET CONTROL MONTH:                                               */  23884108
 /*                                                                  */
```

```
MD.SIG.ACCT.SOURCE(MD61837S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:28:26 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61837S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS MISSING=0;
/*********************************************************************/
/*********************************************************************/
/** THIS PROGRAM COMPARES THE FEEDER FILE AGAINST THE               **/
/** COST SHEET EXPENSE ITEM DATABASE AND THE MASTER KEY OF THE      **/
/** GENERAL LEDGER PROGRAM.  THIS PROGRAM WILL CREATE A SUSPENSE    **/
/** FILE WHICH CONTAINS ANY FEEDER RECORDS WHICH IS NOT FOUND IN    **/
/** OF THE AFOREMENTIONED SYSTEMS.  ALSO, IT WILL PRODUCE A FILE    **/
/** WHICH CONTAINS RECORDS FOUND IN BOTH OF THE AFOREMENTIONED      **/
/** SYSTEMS.                                                        **/
/*********************************************************************/
/*********************************************************************/
/** AUTHOR: MICHAEL C. MENEFIELD                                    **/
/** DATE:   1/27/97                                                 **/
/*********************************************************************/

/*********************************************************************/
/*********************************************************************/
/** THIS MODULE BUILDS THE RECORDS FROM THE COST SHEET EXPENSE ITEM **/
/** DATABASE;                                                       **/

 DATA COSTS ;
   INFILE CSHEET;
     INPUT @01 COSTCTR             $CHAR4.
           @07 EXPCODE             $CHAR4.  ;
     EXPKEY = COSTCTR || EXPCODE;
     DROP COSTCTR EXPCODE;
     OUTPUT;

 PROC SORT DATA=COSTS;
   BY EXPKEY;

 DATA COST1;
   SET COSTS;
   BY EXPKEY;
   IF FIRST.EXPKEY;

/*********************************************************************/


/*********************************************************************/
/*********************************************************************/
/** THIS MODULE BUILTS THE RECORDS FROM THE FEEDER FILE             **/
/**                                                                 **/

  DATA FEEDER1;
  INFILE FEEDER;
  INPUT @001   RECORD1    $CHAR200.
        @201   RECORD2    $CHAR50.
        @001   SOURCE     $CHAR2.
        @004   GLACCT     $CHAR4.
        @009   COSTCTR    $CHAR4.
        @014   EXPCODE    $CHAR4.
        @019   SUBACCT    $CHAR6.
        @026   PROJECT    $CHAR4.
```

```
.SIG.ACCT.SOURCE(MD61838S)
SPRINT 6.1.018   THURSDAY FEBRUARY 27,1997   22:49:15 MDSCMCM  CPU1        **
SPRINT 'MD.SIG.ACCT.SOURCE(MD61838S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
   PAGELEN(60)

PTIONS NONOTES NOSOURCE;
*******************************************************************/
*******************************************************************/
**THIS PROGRAM CREATES THE LISTING REPORTS FOR THE FOLLOWING     **/
**PROCESSES (SOURCES):                                           **/
**  HOURLY PAYROLL, ACCOUNTS PAYABLE, JOB LABOR, SALARY PAYROLL,**/
**  STORES (OPERATING SUPPLIES,SPARES & STORES)                  **/
*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                     **/
/**DATE:   11/15/96                                              **/
/*******************************************************************/
  PROC FORMAT;
       VALUE $TYPEVAL
       'HP' = 'HOURLY PAYROLL'
       'AP' = 'ACCOUNT PAYABLE'
       'JL' = 'JOB LABOR'
       'OP' = 'OPERATING SUPPLIES'
       'SL' = 'SALARY PAYROLL'
       'SP' = 'SPARES'
       'ST' = 'STORES';

/*******************************************************************/
/*******************************************************************/
/* GET ACCOUNTING FILE WHICH IS AN EDIT REPRESENTATION OF THE     */
/*   FOLLOWING FEEDERS FILES:                                     */
/*    HOURLY PAYROLL, SALARY PAYROLL, JOB LABOR, STORES,          */
/*    ACCOUNTS PAYABLE                                            */
/*                                                                */
  DATA FEEDALL;
  INFILE TAPEIN;
  INPUT @001 SOURCE        $CHAR2.
        @004 GLACCT        $CHAR4.
        @009 CENTER        $CHAR4.
        @014 EXPNO         $CHAR4.
        @019 SUBACCT       $CHAR6.
        @026 PROJECT       $CHAR4.
        @036 UNITS         11.2
        @047 DOLLAR        11.2
        @078 SHOP          $CHAR3.
        @159 SLID          $CHAR1.
        @136 WONUM         $CHAR6.
        @205 EFFDT         5.0;
        IF SOURCE = 'AP' THEN WONUM = ' ';
        IF EXPNO = '0000' & SOURCE = 'JL' THEN DELETE;
        OUTPUT FEEDALL;

        IF SOURCE = 'JL' THEN
          DO;
             IF SHOP  = '834' THEN
                SHOP  = '872';
             CENTER = '0'||SHOP;
             GLACCT = '5575';
             EXPNO = '0000';
             DOLLAR = 0;
             SUBACCT = ' ';
```

```
MD.SIG.ACCT.SOURCE(MD61839S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:53:39 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61839S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 /* BATCH */                                                             00010099
 M61839S: PROC OPTIONS(MAIN);                                            00011099
 /********************************************************************/ 00020099
 /********************************************************************/ 00020199
 /* THIS PROGRAM CONVERTS THE INPUT FILE DATA TO THE GENERAL LEDGER **/ 00021099
 /* FILE FORMAT.  THIS PROGRAM WILL USE A CONTROL FILE TO CONVERT   **/ 00022099
 /* THE INPUT FILE TO THE GENERAL LEDGER FORMAT.  THE RECORDS       **/ 00023099
 /* WHICH ARE BUILT FROM THIS PROGRAM ARE IN THE PROPER FORMAT TO   **/ 00023099
 /* BE SENT TO THE GENERAL LEDGER.                                  **/ 00023099
 /********************************************************************/ 00030099
 /********************************************************************/ 00030099
 /**AUTHOR: MICHAEL C. MENEFIELD                                     **/ 00030099
 /**DATE:   11/1/96                                                  **/ 00030099
 /********************************************************************/ 00030099
 /********************************************************************/ 00030099
                                                                         00040099
 /********************************************************************/ 00042099
 /* FILE DECLARATIONS                                                **/ 00042199
 /********************************************************************/ 00043099
  DCL FEEDER  FILE RECORD INPUT;                                         00050099
  DCL VOUCHER FILE RECORD OUTPUT;                                        00052099
  DCL MSGS    FILE RECORD OUTPUT;                                        00052099
 /********************************************************************/ 00060099
 /* MIDDLETOWN WORKS FEEDERS INPUT LAYOUT:                           **/ 00070099
 /********************************************************************/ 00081099
1DCL
  1 FIN_RECORD,                                                          099
    2 FIN_KEY1,                                                          00213099
      3 FIN_SOURCE        CHAR(2)    INIT(' '), /* FEEDER SOURCE    */   00214099
      3 FILLER            CHAR(1)    INIT(' '),                          00214199
      3 FIN_ACCOUNT       CHAR(4)    INIT(' '), /* GEN LEDGER ACCT  */   00214299
      3 FILLER1           CHAR(1)    INIT('-'),                          00214399
      3 FIN_CENTER        CHAR(4)    INIT(' '), /* COST CENTER CODE */   00215099
      3 FILLER2           CHAR(1)    INIT('-'),                          00215199
      3 FIN_EXPCODE       CHAR(4)    INIT(' '), /* EXPENSE CODE     */   00216099
      3 FILLER3           CHAR(1)    INIT('-'),                          00216199
      3 FIN_SUBACCT       CHAR(6)    INIT(' '), /* SUB ACCT CODE    */   00216299
      3 FILLER4           CHAR(1)    INIT('-'),                          00216399
      3 FIN_PROJECT       CHAR(4)    INIT(' '), /* PROJECT NO       */   00216499
      3 FILLER5           CHAR(1)    INIT('-'),                          00216599
      3 FIN_PRODINV       CHAR(4)    INIT(' '), /* PROD INVOICE     */   00216699
      3 FILLER6           CHAR(1)    INIT(' '),                          00216799
    2 FIN_KEY2,                                                          00217299
      3 FIN_UNITS        PIC'(7)S9V.99'  INIT(0),  /* UNITS         */   00217499
      3 FIN_DOLLARS      PIC'(7)S9V.99'  INIT(0),  /* DOLLARS       */   00218099
    2 FIN_KEY3,                                                          00219299
      3 FIN_REGISTER       CHAR(8)     INIT(' '),                        00219399
      3 FILLER7            CHAR(2)     INIT(' '),                        00219499
      3 FIN_COST           CHAR(4)     INIT(' '),                        00219599
      3 FILLER8            CHAR(1)     INIT(' '),                        00219699
      3 FIN_EXP            CHAR(4)     INIT(' '),                        00219799
      3 FILLER9            CHAR(1)     INIT(' '),                        00219899
      3 FIN_INVOICE        CHAR(8)     INIT(' '),                        00219999
      3 FILLER10           CHAR(1)     INIT(' '),                        00220099
      3 FIN_INVDATE        CHAR(7)     INIT(' '),                        00220199
```

```
MD.SIG.ACCT.SOURCE(MD61840S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:16:45 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61840S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS NONOTES NOSOURCE;
/******************************************************************/
/******************************************************************/
/******************************************************************/
/**THIS PROGRAM CREATES THE CONTROL ACCOUNT REPORT FOR THE       **/
/**FOLLOWING PROCESSES (SOURCES):                                **/
/**  HOURLY PAYROLL, ACCOUNTS PAYABLE, JOB LABOR, SALARY PAYROLL,**/
/**  STORES (OPERATING SUPPLIES,SPARES & STORES)                 **/
/**THIS REPORT WILL GIVE THE CREDIT ENTRY TO THE AFOREMENTIONED  **/
/**JOURNEL ENTRIES.                                              **/
/******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                     **/
/**DATE:   11/15/96                                              **/
/******************************************************************/
DATA RECORDA;
  INFILE FILE1;
    INPUT @1  CHARFLAG   $CHAR1. @;
        IF ((CHARFLAG ¬= '-') & (CHARFLAG ¬= 'F'));

    INPUT @1  SOURCE     $CHAR2.
          @6  ACCT       $CHAR4.
          @11 CNTR       $CHAR4.
          @16 EXPN       $CHAR4.
          @21 SUBT       $CHAR4.
          @31 PROD       $CHAR4.
          @37 SNAME      $CHAR6.;

          EXPKEY = SNAME||ACCT||CNTR||EXPN||SUBT;
    DROP CHARFLAG;

PROC SORT DATA=RECORDA;
  BY EXPKEY;

DATA RECORDB;
  SET RECORDA;
  BY EXPKEY;
  IF FIRST.EXPKEY;

PROC SORT DATA=RECORDB;
  BY EXPKEY;

DATA RECORD2;
  INFILE FILE2;
    INPUT @09   SNAME      $CHAR6.
          @79   ACCT       $CHAR4.
          @83   CNTR       $CHAR4.
          @87   EXPN       $CHAR4.
          @91   SUBT       $CHAR4.
          @97   PROD       $CHAR4.
          @288  DOLLAR     PD8.2
          @296  UNIT       PD8.2;

          EXPKEY =  SNAME||ACCT||CNTR||EXPN||SUBT;

PROC SORT DATA=RECORD2;
```

```
MD.SIG.ACCT.SOURCE(MD61841S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:25:32 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61841S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 /*BATCH  */                                                           00010004
 M61841S:  PROC (TP_PCB_PTR,MDCSTSHT_PTR,MDCSTSH2_PTR,
                 MDEXPITM_PTR)
           OPTIONS (MAIN) REORDER;                                     23872008
 /*********************************************************************/23879300
 /* THIS PROGRAM SEARCHES THE COST SHEETS DATABASE FOR THE OPEN       */23879408
 /*   EXPENSE ITEMS IN THE DATABASE.                                  */23879408
 /*   THIS PROGRAM BUILDS A FILE THAT CONTAINS VALID ACCOUNT CODES    */23879408
 /*   AND EXPENSE CODES.                                              */23879408
 /*********************************************************************/23879508
 /*********************************************************************/23879508
 /** AUTHOR: MICHAEL C. MENEFIELD                                     **/23879508
 /** DATE:                                                            **/23879508
 /*********************************************************************/23879508
-DEFAULT RANGE(*) STATIC;                                              23879600
 DCL  PLITDLI                       EXTERNAL ENTRY;                    23879702
-DCL  TP_PCB_PTR                       POINTER;                        23879802
-DCL  MDCSTSH2_PTR                     POINTER;                        23879802
-DCL  MDCSTSHT_PTR                     POINTER;                        23879802
-DCL  MDEXPITM_PTR                     POINTER;                        23879802
                                                                       23880008
 DCL  1 MDCSTSH2                   BASED (MDCSTSH2_PTR),               23880908
 %INCLUDE DBPCB;
                                   CHAR(8);

 DCL  1 MDCSTSHT                   BASED (MDCSTSHT_PTR),               23880908
 %INCLUDE DBPCB;
                                   CHAR(4);

 DCL  1 MDEXPITM                   BASED (MDEXPITM_PTR),               23880908
 %INCLUDE DBPCB;
                                    CHAR(11);
1%INCLUDE TPPCB;                                                       23880800
1%INCLUDE EXPITM;
1%INCLUDE IMSCON;                                                      23884000

 DCL  ADDR                     BUILTIN;

 DCL  PATH_IO3                 CHAR (990);
 DCL  MSGS                          FILE RECORD OUTPUT;                23880200
 DCL  CSDATA                        FILE RECORD OUTPUT;                23880200

0DCL  MSG_IO                        CHAR (133);                        23883200
 DCL  1 DATABASE_IO,
        2 DATABASE_COSTEXPN$      CHAR(11) INIT(' '),
        2 FILLER                  CHAR(69) INIT(' ');

 DCL  MONTH                        PIC '99' INIT(0);
 DCL  EOF                           BIT (1)  INIT('0'B);               23884300
 DCL  ON                            BIT (1)  INIT('1'B);               23884400
 DCL  OFF                           BIT (1)  INIT('0'B);               23884500

1/*********************************************************************/ 23884108
 /********************************************************************/ 23884108
 /* INITIALIZE : MAIN PROCEDURE                                      */ 23884108
```

```
MD.SIG.ACCT.SOURCE(MD61842S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:00:38 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61842S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 M61842S: PROC OPTIONS (MAIN) REORDER;
 /*********************************************************************/23879300
 /*********************************************************************/23879300
 /* THIS PROGRAM WILL SEPARATE THE JOB LABOR RECORD BASED ON BUSINESS */23879408
 /* MONTH.  THIS ENTAILS BUILDING A FILE WITH THE CURRENT BUSINESS    */23879408
 /* MONTH RECORDS AND A SUSPENSE FILE CONTAINING OTHER BUSINESS MONTH */23879408
 /* RECORDS.                                                          */23879508
 /*********************************************************************/23879508
 /*********************************************************************/23879508
 /** AUTHOR: MICHAEL C. MENEFIELD                                     **/23879508
 /** DATE:  7/31/96                                                   **/23879508
 /*********************************************************************/23879508
 /*********************************************************************/23879508

 DCL  JOBLABR                         FILE RECORD INPUT;                23880200
 DCL  MATCH                           FILE RECORD OUTPUT;               23880200

 DCL  1 RAW_RECORD_IO,
        2 FILLER1   CHAR(36)  INIT(' '),
        2 RAW_MONTH CHAR(2)   INIT(' '),
        2 FILLER2   CHAR(14)  INIT(' ');

  DCL OFF           BIT(1)          INIT('0'B);
  DCL ON            BIT(1)          INIT('1'B);
  DCL EOF_JOBLABOR  BIT(1)          INIT('0'B);
  DCL MONTH         CHAR(2)         INIT(' ');

1/*******************************************************************/   23884108
 /*******************************************************************/   23884108
 /* INITIALIZE : MAIN PROCEDURE                                     */   23884108
 /*******************************************************************/   23884108
 ON ENDFILE(JOBLABR) EOF_JOBLABOR = ON;
 CALL GET_CONTROL_MONTH;
 DO WHILE (¬EOF_JOBLABOR);
   READ FILE (JOBLABR) INTO (RAW_RECORD_IO);
   IF ((¬EOF_JOBLABOR) & (RAW_MONTH = MONTH)) THEN
        WRITE FILE (MATCH)   FROM (RAW_RECORD_IO);
 END;

1/*******************************************************************/   23884108
 /*******************************************************************/   23884108
 /* GET CONTROL MONTH:                                              */   23884108
 /*                                                                 */
  GET_CONTROL_MONTH: PROC;
    DCL  CNTLCRD                         FILE RECORD INPUT;             23880200
    DCL 1 CNTLCARD_IO,
          2 CNTLCARD_MONTH CHAR(3),
          2 FILLER         CHAR(77);

   READ FILE (CNTLCRD) INTO (CNTLCARD_IO);
   SELECT(CNTLCARD_MONTH);
     WHEN('JAN') MONTH = '01';
     WHEN('FEB') MONTH = '02';
     WHEN('MAR') MONTH = '03';
     WHEN('APR') MONTH = '04';
```

```
.SIG.ACCT.SOURCE(MD61843S)
'SPRINT 6.1.018   FRIDAY FEBRUARY 28,1997         5:41 MDSCMCM  CPU1        **
'SPRINT 'MD.SIG.ACCT.SOURCE(MD61843S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
   PAGELEN(60)

ᐣTIONS;
*********************************************************************/
*********************************************************************/
*  MD61843S:  THIS PROGRAM JOINS TOGETHER THE RATES FILE, SHOP      */
*             NAME AND DESCRIPTION FILE WITH THE RAW JOB LABOR      */
*             FILE.  THIS FILE WILL BE THE INPUT FILE FOR THE       */
*             JOB LABOR PROCESS.                                    */
'********************************************************************/
'********************************************************************/
/** AUTHOR: MICHAEL C. MENEFIELD                                    **/
/** DATE:   8/23/96                                                 **/
/*********************************************************************/

/************ READ IN SHOP RATES *****************/
/************************************************/
  DATA SHOPRATE;
    INFILE LBORRATE;
    INPUT @31 SHOP     $CHAR3.
          @34 RATE      4.2;

/******* READ IN WORK ORDER INCLUDING COST CENTER CODE ******/
/**********************************************************/
DATA SHOPHRS;
 INFILE JLACCSEQ;
 INPUT @2  PLANT             1.0
       @3  SHOP              $CHAR3.
       @6  HOUR              3.1
       @9  WORK              $CHAR6.
       @15 SEQNO             $CHAR2.
       @18 YWORK             $CHAR2.
       @20 RECPLANT          1.0
       @21 TYPE1            $CHAR1.
       @22 COST1            $CHAR4.
       @26 COST2            $CHAR4.
       @30 COST3            $CHAR4.
       @37 BATCH            5.0
       @42 SEQCOUNT         5.0
       @47 BADGE            5.0;

       TOTWORK = YWORK||WORK;

       IF TYPE1 = 'Z' THEN SELFFLAG = 'Y';
       IF SUBSTR(COST1,1,1) = '-' THEN
           DO
                TYPE1 = ' ';
                COST1 = '0000';
                SELFFLAG = 'Y';
           END;
       ELSE
           SELFFLAG = 'N';
       IF SUBSTR(COST2,1,1) = '-' THEN COST2 = '0000';
       IF SUBSTR(COST3,1,1) = '-' THEN COST3 = '0000';
       COSTCNTR = COST1||' '||COST2||' '||COST3;
       DROP COST1 COST2 COST3;
***;
```

```
MD.SIG.ACCT.SOURCE(MD61844S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:56:55 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61844S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 M61844S: PROC OPTIONS (MAIN) REORDER;
 /*********************************************************************/23879300
 /*********************************************************************/23879300
 /* THIS PROGRAM WILL EDIT THE RAW JOB LABOR RECORDS TO CONVERT       */23879408
 /* 'S' SYMBOLS TO A SELF-HOUR ACCOUNT CODES AND IT WILL PLACE THE    */23879408
 /* DATE OF THE CURRENT MONTH INTO THE RECORD.                        */23879408
 /*********************************************************************/23879508
 /*********************************************************************/23879508
 /** AUTHOR: MICHAEL C. MENEFIELD                                    **/23879508
 /** DATE:   8/26/96                                                 **/23879508
 /*********************************************************************/23879508
 /*********************************************************************/23879508

 DCL  JOBLABR                      FILE RECORD INPUT;                  23880200
 DCL  JOBEDIT                      FILE RECORD OUTPUT;                 23880200
 DCL  SUBSTR          BUILTIN;
 DCL  1 RAW_RECORD_IO,
        2 FILLER1    CHAR(20)  INIT(' '),
        2 TIPE       CHAR(1)   INIT(' '),
        2 COSTCTR    CHAR(4)   INIT(' '),
        2 FILLER2    CHAR(11)  INIT(' '),
        2 BATCH      CHAR(2)   INIT(' '),
        2 FILLER3    CHAR(14)  INIT(' ');

  DCL OFF            BIT(1)          INIT('0'B);
  DCL ON             BIT(1)          INIT('1'B);
  DCL EOF_JOBLABOR   BIT(1)          INIT('0'B);
  DCL MONTH          CHAR(2)         INIT(' ');
  DCL FIRST_CHAR     CHAR(1)         INIT(' ');

1/********************************************************************/  23884108
 /********************************************************************/  23884108
 /* INITIALIZE : MAIN PROCEDURE                                      */  23884108
 /********************************************************************/  23884108
 ON ENDFILE(JOBLABR) EOF_JOBLABOR = ON;
 CALL GET_CONTROL_MONTH;
 DO WHILE (¬EOF_JOBLABOR);
   READ FILE (JOBLABR) INTO (RAW_RECORD_IO);
   IF (¬EOF_JOBLABOR) THEN
     DO;
       IF (MONTH ¬= ' ') THEN BATCH = MONTH;

       FIRST_CHAR = SUBSTR(COSTCTR,1,1);

       SELECT(FIRST_CHAR);
         WHEN('-') TIPE = '-';
         WHEN('8') TIPE = 'S';
         OTHERWISE;
       END;

       WRITE FILE (JOBEDIT) FROM (RAW_RECORD_IO);
    END;
 END;

1/********************************************************************/  23884108
```

```
MD.SIG.ACCT.SOURCE(MD61847S)
VPSPRINT 6.1.018   FRIDAY FEBRUARY 28,1997        21:34 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61847S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

************************************************************************ 00010099
** THE PURPOSE OF THIS PROGRAM IS TO CREATE THE MONTHLY ROLL-UP       ** 00020099
** REPORT.  THIS REPORT WILL CONTAINS THE MONTH-TO-DATE DATA OF        ** 00030099
** JOB LABOR SORT BY DEPARTMENT AND COST CENTER.                      ** 00040099
************************************************************************ 00050099
************************************************************************ 00060099
** AUTHOR: MICHAEL C. MENEFIELD                                       ** 00070099
** DATE:   2/26/96                                                    ** 00080099
************************************************************************ 00090099
************************************************************************ 00100099
OPTIONS;                                                                 00110000
DATA RAWAPPR;                                                            00120030
  INFILE APPROP;                                                         00130087
  INPUT @5  ORDERNO  $CHAR4.                                             00140099
        @9  CHGTO    $CHAR4.;                                            00150065
                                                                         00160029
PROC SORT DATA=RAWAPPR;                                                  00170030
  BY ORDERNO;                                                            00180030
                                                                         00190003
DATA DATABASE;                                                           00200030
  INFILE CSTSHT;                                                         00210088
  INPUT @1  COSTCTR  $CHAR4.                                             00220030
        @6  DEPTNO   $CHAR2.                                             00230030
        @9  COSTNAME $CHAR20.;                                           00240030
                                                                         00250030
PROC SORT DATA=DATABASE;                                                 00260030
  BY COSTCTR;                                                            00270030
                                                                         00280056
DATA JOBLABOR;                                                           00290005
  FORMAT ORDERNO $CHAR4.;                                                00300099
  INFILE JOBLBOR;                                                        00310087
  INPUT @09  COSTCTR  $CHAR4.                                            00320064
        @26  ORDERNO  $CHAR4.                                            00330099
        @36  HOURS    11.2                                               00340059
        @47  DOLLARS  11.2                                               00350006
        @78  SHOP     $CHAR3.                                            00360006
        @95  SHOPNAME $CHAR25.                                           00370006
        @136 WONUM    $CHAR6.                                            00380006
        @152 SHOPONLY $CHAR1.                                            00390099
        @172 MDAY     2.0                                                00400086
        @174 DDAY     2.0                                                00410086
        @212 DESCRIP  $CHAR39.;                                          00420086
                                                                         00430099
   IF SHOPONLY = 'N';                                                    00440099
   IF MDAY ¬= MONTH THEN                                                 00450086
      DO;                                                                00460086
          MONTH = MDAY;                                                  00470086
          DAYS  = 0;                                                     00480086
          RETAIN MONTH DAYS;                                             00490086
          CALL SYMPUT('HEADMTH',MONTH);                                  00500086
      END;                                                               00510086
   IF DDAY > DAYS THEN                                                   00520086
      DO;                                                                00530086
         DAYS = DDAY;                                                    00540086
         RETAIN DAYS;                                                    00550086
```

```
MD.SIG.ACCT.SOURCE(MD61849S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      19:16:26 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61849S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS MISSING = 0;                                                    00010015
/********************************************************************/  00020030
/********************************************************************/  00030030
/* THIS PROGRAM CREATES THE JOB LABOR SHOP HOURS REPORT.           **/  00040030
/********************************************************************/  00050030
/********************************************************************/  00060030
/**AUTHOR: MICHAEL C. MENEFIELD                                      */  00070030
/**DATE:   8/26/96                                                   */  00080030
/********************************************************************/  00090030
DATA SOURCE;                                                            00100000
  INFILE JOBLBOR;                                                       00110020
  INPUT @36  HOUR      11.2                                             00120001
        @47  DOLLAR    11.2                                             00130001
        @78  SHOP      $CHAR3.                                          00140000
        @95  SHOPNAME  $CHAR25.                                         00150000
        @126 RATE      5.2                                              00160000
        @152 SHOPONLY  $CHAR1.                                          00170000
        @158 OTHERCHG  $CHAR1.                                          00180017
        @205 EFFDT     5.0;                                             00190017
                                                                        00200017
      /**************************************/                          00210017
      /**************************************/                          00220017
      IF GETDATE ¬= 1 THEN                                              00230017
        DO;                                                             00240017
          YEAR = INT(EFFDT / 365.25);                                   00250017
          DAY  = EFFDT - (YEAR * 365.25) +1;                            00260017
          CALDAY = YEAR * 1000 + DAY;                                   00270017
                                                                        00280017
          JULIAN= CALDAY;                                               00290017
          SASDATE = DATEJUL(JULIAN);                                    00300017
          DATE    = PUT(SASDATE,MMDDYY8.);                              00310019
                                                                        00320017
          GETDATE = 1;                                                  00330017
          RETAIN GETDATE;                                               00340017
          CALL SYMPUT('DHEADER',DATE);                                  00350017
        END;                                                            00360017
      /**************************************/                          00370017
      /**************************************/                          00380017
                                                                        00390017
                                                                        00400000
DATA SOURCE2;                                                           00410000
  SET SOURCE;                                                           00420000
  SELECT;                                                               00430005
  WHEN(OTHERCHG = 'A' | OTHERCHG = 'C')                                 00440000
     DO;                                                                00450000
         SHOUR   = HOUR;                                                00460000
         DHOUR   = HOUR;                                                00470000
         SDOLLAR = DOLLAR;                                              00480000
     END;                                                               00490000
  WHEN(SHOPONLY = 'Y')                                                  00500023
     DO;                                                                00510023
       DOLLAR = 0;                                                      00520023
       SELFHOUR = HOUR;                                                 00530023
     END;                                                               00540000
  OTHERWISE                                                             00550000
     DO;
```

```
MD.SIG.ACCT.SOURCE(MD61851S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      19:21:27 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61851S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS;
/********************************************************************/
/********************************************************************/
/* THE PURPOSE OF THIS PROGRAM IS TO SUMMARIZE THE JOB LABOR       */
/* RECORDS BY COST CENTER, EXPENSE CODE AND WORK ORDER NUMBER.     */
/********************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                       **/
/**DATE:   08/01/96                                                **/
/********************************************************************/
 DATA SOURCE;
 INFILE RAWJOB;
 INPUT @1   SOURCE   $CHAR2.
       @4   GLACCT   $CHAR4.
       @9   COSTCTR  $CHAR4.
       @14  EXPNCODE $CHAR4.
       @19  SUBACCT  $CHAR6.
       @26  PROJECT  $CHAR4.
       @31  PRODINV  $CHAR4.
       @36  UNIT     11.2
       @47  DOLLARS  11.2
       @159 SLID     $CHAR1.
       @182 SLAC1    $CHAR4.
       @186 SLAC2    $CHAR4.
       @190 MEASURE  $CHAR2.
       @205 CENTDATE $CHAR5.;

 PROC SORT DATA=SOURCE;
   BY SOURCE  GLACCT  COSTCTR EXPNCODE SUBACCT PROJECT
      PRODINV MEASURE SLID SLAC2 SLAC1;

 PROC MEANS DATA=SOURCE NOPRINT MAXDEC=2;
   ID  CENTDATE;
   BY  SOURCE  GLACCT  COSTCTR EXPNCODE SUBACCT PROJECT
       PRODINV MEASURE SLID SLAC2 SLAC1;
   VAR UNIT DOLLARS;
   OUTPUT OUT=NEWFILE SUM= UNIT DOLLARS;

 PROC PRINT;
    SUM UNIT DOLLARS;

 DATA _NULL_;
 SET NEWFILE;
   FILE SUMJOB;
   PUT @1   SOURCE   $CHAR2.
       @4   GLACCT   $CHAR4.
       @9   COSTCTR  $CHAR4.
       @14  EXPNCODE $CHAR4.
       @19  SUBACCT  $CHAR6.
       @26  PROJECT  $CHAR4.
       @31  PRODINV  $CHAR4.
       @36  UNIT     11.2
       @47  DOLLARS  11.2
       @159 SLID     $CHAR1.
       @182 SLAC1    $CHAR4.
       @186 SLAC2    $CHAR4.
```

```
MD.SIG.ACCT.SOURCE(MD61854S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      19:09:47 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61854S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 M61854S: PROC OPTIONS (MAIN) REORDER;
 /***********************************************************************/23879300
 /***********************************************************************/23879300
 /** THIS PROGRAM COMBINES THE NEW PURCHASE ORDER NUMBER AND           */23879408
 /** DESCRIPTION WITH THE ACCOUNTS PAYABLE FILES.                      */ 23879408
 /***********************************************************************/23879300
 /***********************************************************************/23879300
 /*AUTHOR: MICHAEL C. MENEFIELD                                        */23879300
 /*DATE:   10/31/96                                                    */23879300
 /***********************************************************************/23879300

 DCL  ACCTPAY                   FILE RECORD INPUT;                      23880200
 DCL  APDESC                    FILE RECORD OUTPUT;                     23880200
 DCL  PORDER FILE RECORD INPUT KEYED ENV(VSAM);                         23880200
 DCL  CORDER FILE RECORD INPUT KEYED ENV(VSAM);                         23880200

 DCL  MSG_IO                     CHAR(133) INIT(' ');

 DCL 1 ACCTPAY_RECORD,
       2 ACCTPAY_IO,
         3 IN_TYPE            CHAR(1)         INIT(' '),
         3 FILLER1            CHAR(72)        INIT(' '),
         3 IN_PURCHASE        CHAR(11)        INIT(' '),
         3 FILLER2            CHAR(96)        INIT(' '),
       2 DESCRIPTION          CHAR(40)        INIT(' ');

 DCL   1 RECORD_IO,
         2 PNUMBER                CHAR(11) INIT(' '),
         2 DESCRIPTION            CHAR(40) INIT(' '),
         2 FILLER1                CHAR(1)  INIT(' ');

 DCL OFF            BIT(1)          INIT('0'B);
 DCL ON             BIT(1)          INIT('1'B);
 DCL EOF_ACCTPAY    BIT(1)          INIT('0'B);
 DCL KEY_FOUND      BIT(1)          INIT('0'B);
1/*********************************************************************/  23884108
 /*********************************************************************/  23884108
 /** INITIALIZE : MAIN PROCEDURE                                     **/  23884108
 /**                                                                 **/  23884108
 ON ENDFILE(ACCTPAY) EOF_ACCTPAY = ON;
 ON KEY (PORDER) BEGIN;
                    KEY_FOUND = OFF;
                 END;
 ON KEY (CORDER) BEGIN;
                    KEY_FOUND = OFF;
                 END;

 DO WHILE (¬EOF_ACCTPAY);
   ACCTPAY_RECORD.DESCRIPTION = ' ';
   KEY_FOUND  = ON;

   READ FILE (ACCTPAY) INTO (ACCTPAY_IO);
   SELECT(IN_TYPE);
     WHEN('H');
```

```
MD.SIG.ACCT.SOURCE(MD61855S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:20:38 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61855S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

 /* BATCH */
 M61855S: PROC OPTIONS(MAIN);
 /********************************************************************/
 /********************************************************************/
 /* THIS PROGRAM CONVERTS THE INPUT FILE DATA TO THE GENERAL LEDGER **/
 /* FILE FORMAT.  THIS PROGRAM WILL USE A CONTROL FILE TO           **/
 /* PROVIDE THE LOGIC TO CONVERT THE INPUT DATA INTO USABLE         **/
 /* ACCOUNTING CODES, COST CENTER, EXPENSE ITEMS, SUB ACCOUNTS,     **/
 /* PRODUCT CODES AND PROJECT CODES.                                **/
 /********************************************************************/
 /*AUTHOR: MICHAEL C. MENEFIELD                                     **/
 /*DATE:   10/31/96                                                 **/
 /********************************************************************/
 /********************************************************************/

 /********************************************************************/
 /* FILE DECLARATIONS                                               **/
 /********************************************************************/
  DCL FEEDER  FILE RECORD INPUT;
  DCL FEEDERS FILE RECORD OUTPUT;
  DCL MSGS    FILE RECORD OUTPUT;
 /********************************************************************/
 /* MIDDLETOWN WORKS FEEDERS INPUT LAYOUT:                          **/
 /********************************************************************/
  DCL 1 FEEDER_IN,
        2 IN_TYPE          CHAR(1)         INIT(' '),
        2 IN_REGISTER      CHAR(8)         INIT(' '),
        2 IN_DUNS          CHAR(9)         INIT(' '),
        2 IN_VENDOR        CHAR(30)        INIT(' '),
        2 IN_INVOICE       CHAR(8)         INIT(' '),
        2 IN_INVOICE_YEAR  PIC '99'        INIT(0),
        2 IN_INVOICE_DAYS  PIC '999'       INIT(0),
        2 FILLER2          CHAR(6)         INIT(' '),
        2 IN_SALETAX       FIXED DEC(7,2)  INIT(0),
        2 FILLER3          CHAR(2)         INIT(' '),
        2 IN_PURCHASE      CHAR(15)        INIT(' '),
        2 IN_NETDOLL       FIXED DEC(11,2) INIT(0),
        2 IN_UNIT          FIXED DEC(11,1) INIT(0),
        2 FILLER4          CHAR(13)        INIT(' '),
        2 LEDGER_CODE      CHAR(1)         INIT(' '),
        2 COST_CENTER      CHAR(4)         INIT(' '),
        2 EXPENSE_CODE     CHAR(4)         INIT(' '),
        2 SUBACCT_CODE     CHAR(4)         INIT(' '),
        2 FILLER5          CHAR(5)         INIT(' '),
        2 IN_SRN           CHAR(7)         INIT(' '),
        2 FILLER7          CHAR(3)         INIT(' '),
        2 IN_PLANT         CHAR(2)         INIT(' '),
        2 FILLER8          CHAR(37)        INIT(' '),
        2 IN_DESCRIPTION   CHAR(39)        INIT(' '),
        2 FILLER9          CHAR(1)         INIT(' ');
 /********************************************************************/
 /* GENERAL LEDGER RECORD LAYOUT                                    **/
 /********************************************************************/
1DCL
  1 FDR_FILE,
MD.SIG.ACCT.SOURCE(MD61861S)
VPSPRINT 6.1.018   WEDNESDAY MARCH 5,1997      18:23:27 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61861S)' MDV7401U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS NONOTES NOSOURCE;
/*******************************************************************/
/*******************************************************************/
/**THIS PURPOSE OF THIS PROGRAM IS COLLECT ALL GENERAL LEDGER    **/
/**ACCOUNT CODES THAT ARE EQUAL TO 1430.  THIS PROGRAM WILL      **/
/**CREATE A FILE WHICH CONTAINS ONLY 1430 ACCOUNTS.              **/
/*******************************************************************/
/**AUTHOR: MICHAEL MENEFIELD                                     **/
/**DATE:   11/15/96                                              **/
/*******************************************************************/

/*******************************************************************/
/*******************************************************************/
/* SUMMERIZE THE JOB LABOR DETAIL RECORD                           */
/*                                                                 */
  DATA ONLY1430;
  INFILE FEEDER;
  INPUT @001 RECORDA       $CHAR200.
        @201 RECORDB       $CHAR50.
        @001 SOURCE        $CHAR2.
        @004 GLACCT        $CHAR4.
        @009 CENTER        $CHAR4.;

  IF GLACCT = '1430';
  IF CENTER ¬= '1440' & CENTER ¬= '1441';

DATA _NULL_;
  SET ONLY1430;
  FILE FILE1430;
    PUT @001 RECORDA       $CHAR200.
        @200 RECORDB       $CHAR50.;
```

```
.SIG.ACCT.SOURCE(MD61862S)
SPRINT 6.1.018   THURSDAY FEBRUARY 27,1997   22:59:09 MDSCMCM  CPU1       **
SPRINT 'MD.SIG.ACCT.SOURCE(MD61862S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
  PAGELEN(60)

'TIONS NONOTES NOSOURCE;
******************************************************************/
******************************************************************/
* THE PURPOSE OF THIS PROGRAM IS TO CREATE A REPORT TO SUMMARIZE */
* THE UNIT AND DOLLARS BY COST CENTER AND EXPENSE NUMBER         */
******************************************************************/
**AUTHOR: MICHAEL MENEFIELD                                     **/
**DATE:   11/19/96                                              **/
******************************************************************/

 PROC FORMAT;
      VALUE $TYPEVAL
      'HP' = 'HOURLY PAYROLL'
      'AP' = 'ACCOUNT PAYABLE'
      'JL' = 'JOB LABOR'
      'OP' = 'OPERATING SUPPLIES'
      'SL' = 'SALARY PAYROLL'
      'SP' = 'SPARES'
      'ST' = 'STORES';

/*****************************************************************/
/*****************************************************************/
/* GET ACCOUNTING FILE WHICH IS AN EDIT REPRESENTATION OF THE     */
/*   FOLLOWING FEEDERS FILES:                                     */
/*    HOURLY PAYROLL, SALARY PAYROLL, JOB LABOR, STORES,          */
/*    ACCOUNTS PAYABLE                                            */
/*                                                                */
  DATA FEEDALL;
  INFILE TAPEIN;
  INPUT @001 SOURCE        $CHAR2.
        @004 GLACCT        $CHAR4.
        @009 CENTER        $CHAR4.
        @014 EXPNO         $CHAR4.
        @019 SUBACCT       $CHAR6.
        @026 PROJECT       $CHAR4.
        @036 UNITS         11.2
        @047 DOLLAR        11.2
        @078 SHOP          $CHAR3.
        @159 SLID          $CHAR1.
        @136 WONUM         $CHAR6.
        @205 EFFDT         5.0;
        IF EXPNO ¬= '0000';

        IF GLACCT ¬= '5575' & GLACCT ¬= '1920' THEN
           CENTER = '****';
        IF GLACCT = '1920' THEN CENTER = PROJECT;

  PROC SORT DATA=FEEDALL;
    BY SOURCE CENTER EXPNO WONUM;

  PROC MEANS DATA=FEEDALL NOPRINT MAXDEC=2;
    ID EFFDT;
    BY SOURCE CENTER EXPNO;
    VAR UNITS DOLLAR;
    OUTPUT OUT=SUMFEED SUM=UNITS DOLLAR;
```

```
MD.SIG.ACCT.SOURCE(MD61867S)
VPSPRINT 6.1.018   THURSDAY FEBRUARY 27,1997   22:56:48 MDSCMCM  CPU1       **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61867S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

OPTIONS;
/**********************************************************************/
/**********************************************************************/
/**  ACCOUNTS PAYABLE FILE 560 PRINT PROGRAM                         **/
/**  THIS REPORT PRINTS A FICHE COPY EACH TIME IT PRINTS             **/
/**  SO BE CAREFUL WHEN COPYING OR PRINTING                          **/
/**********************************************************************/
 DATA APAYFILE;
 INFILE NEWAPF;
 INPUT   @001  SOURCE       $CHAR2.
         @004  GLACCT       $CHAR4.
         @009  CENTER       $CHAR4.
         @014  EXPNSNO      $CHAR4.
         @019  SUBACCT      $CHAR6.
         @026  ORDERNO      $CHAR4.
         @031  PRODINV      $CHAR4.
         @036  UNITS        11.1
         @047  DOLLAR       11.2
         @059  REGISTER     $CHAR8.
         @068  COST         $CHAR4.
         @073  EXP          $CHAR4.
         @073  HEXP         $CHAR4.
         @078  INVOICE      $CHAR8.
         @087  INVDATE      DATE7.
         @095  VENDOR       $CHAR21.
         @126  DUNS         $CHAR9.
         @136  PURCHASE     $CHAR15.
         @152  BATCH        $CHAR2.
         @155  DEPT         $CHAR2.
         @158  PREFIX       $CHAR1.
         @160  CHARGE       $CHAR30.
         @165  CHRICHG1     $CHAR4.
         @169  CHRICHG2     $CHAR4.
         @192  HMONTH       $CHAR2.
         @195  HDAY         $CHAR2.;

   DATA APAYFIL2;
   INFILE NEWAPF2;
   INPUT   @001  SOURCE       $CHAR2.
           @004  GLACCT       $CHAR4.
           @009  CENTER       $CHAR4.
           @014  EXPNSNO      $CHAR4.
           @019  SUBACCT      $CHAR6.
           @026  ORDERNO      $CHAR4.
           @031  PRODINV      $CHAR4.
           @036  UNITS        11.1
           @047  DOLLAR       11.2
           @059  REGISTER     $CHAR8.
           @068  COST         $CHAR4.
           @073  EXP          $CHAR4.
           @073  HEXP         $CHAR4.
           @078  INVOICE      $CHAR8.
           @087  INVDATE      DATE7.
           @095  VENDOR       $CHAR21.
           @126  DUNS         $CHAR9.
```

```
MD.SIG.ACCT.SOURCE(MD61871S)
VPSPRINT 6.1.018   THURSDAY FEBRUARY 27,1997   22:48:56 MDSCMCM  CPU1        **
VPSPRINT 'MD.SIG.ACCT.SOURCE(MD61871S)' MDV7403U CLASS(A)  FORM(PORT) COPIES(1)
    PAGELEN(60)

**********************************************************************
**THIS PROGRAM IS TO CONVERT THE SIX DIGIT BURDEN EXPENSE ITEM TO    **
**A FOUR DIGIT EXPENSE ITEM.  THE NEWLY CONVERTED EXPENSE ITEMS WILL **
**BEGIN WITH A '6'.                                                  **
**********************************************************************
**********************************************************************
**AUTHOR: MICHAEL C. MENEFIELD                                       **
**DATE:   JANUARY 31, 1996                                           **
**********************************************************************
**********************************************************************
OPTIONS;
DATA CSTSHT1;
  INFILE CSTSHT;
  INPUT @1 CENTER  $CHAR4.
        @5 EXPNITM $CHAR6.;

PROC SORT DATA=CSTSHT1;
  BY EXPNITM;

DATA BURDENS;
  INFILE RAWEXPN;
  INPUT @1 NEWEXP  $CHAR4.
        @6 EXPNITM $CHAR6.;

PROC SORT DATA=BURDENS;
  BY EXPNITM;

DATA NEWEXPN;
  MERGE CSTSHT1 (IN=X)
        BURDENS (IN=Y);
  BY EXPNITM;
  IF X;
  FORMAT EXPENSE $CHAR6.;
  EXPENSE = EXPNITM;
  IF X=Y THEN
     DO;
         SUBSTR(EXPENSE,1,2) = 99;
         SUBSTR(EXPENSE,3,4) = NEWEXP;
     END;

PROC SORT DATA=NEWEXPN;
   BY CENTER EXPENSE;

DATA _NULL_;
  SET NEWEXPN;
  FILE CHGFILE;
  PUT @1 CENTER  $CHAR4.
      @5 EXPENSE $CHAR6.;
```