

# **DESIGN AND IMPLEMENTATION OF A GENERAL LEDGER ACCOUNTING MANAGEMENT SYSTEM**

**A Case Study of Alu – Diz Limited**

**Danbaba Street Kano, Kano State**

**By**

**OLANREWAJU, ADEBAYO MOSES**

***PGD/MCS/98/1999/0852***

**DEPT. OF MATH/COMPUTER SCIENCE  
FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.  
NIGER STATE.**

***SEPTEMBER 2001.***

## CERTIFICATION

We agree in every capacity to certify that this project work was produced by the bearer  
**OLANREWAJU, ADEBAYO MOSES** in partial fulfillment of the Post Graduate  
Course in Computer Science.

---

Mr. Audu Isah  
*Supervisor*

---

Date

---

Dr. S. A. Reju  
*Head of Department*

---

Date

---

External Examiner

---

Date

## TABLE OF CONTENTS

Title Page	-	-	-	-	i
Certification	-	-	-	-	ii
Dedication	-	-	-	-	iii
Acknowledgement	-	-	-	-	iv
Table of Contents	-	-	-	-	v
Abstract					

## CHAPTER ONE: INTRODUCTION

1.1	Background Study	-	-	-	-	1
1.2	Project Objective	-	-	-	-	1
1.3	Methodology	-	-	-	-	1
1.4	Problem Definition	-	-	-	-	2
1.5	Feasibility Study	-	-	-	-	2
1.6	Facts Finding Technique	-	-	-	-	2
1.7	Scope and Limitation of Study	-	-	-	-	3

## CHAPTER TWO: LITERATURE REVIEW

2.1	Historical Study – General Bookkeeping Principles	-	-	-	-	7
2.2	Accounting Information	-	-	-	-	8
2.3	Accounting principles	-	-	-	-	10
2.4	Benefits For An Automated Bookkeeping and General Ledger System-					15
2.5	Underlining Concepts of Automated Systems	-	-	-	-	16

## CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN- 18

3.1	Design Strategies For Implementing Computer Based Ledger System	-				18
3.2	Planning System Requirement	-	-	-	-	19
3.3	Potential Hurdles & Solution	-	-	-	-	19

3.4	System Flowchart	-	-	-	22
3.5	Input Specification & Output Specification-	-	-	-	24
3.6	Input Design & Output Design	-	-	-	28

#### CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.1	Introduction	-	-	-	53
4.2	Requirements For Implementing A Computerized Ledger System	-	-	-	53
4.2.1	Hardware Requirement-	-	-	-	54
4.2.2	Software Requirement-	-	-	-	55
4.2.3	Procedure	-	-	-	56
4.2.4	Human Resources	-	-	-	57
4.3	System Testing & Setup-	-	-	-	57
4.4	Change-Over	-	-	-	58

#### CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

4.1	Conclusion	-	-	-	61
4.2	Recommendation	-	-	-	61

<b>References</b>	-	-	-	-	<b>62</b>
-------------------	---	---	---	---	-----------

#### Appendix

-	Program Source Codes	-	-	-	a - o
---	----------------------	---	---	---	-------

## **CHAPTER ONE**

### **1.1 BACKGROUND OF STUDY**

Accounting as it exists today may be viewed as a system of assumptions, doctrines, tenets, and conventions, all encompassed by the phrase "generally accepted accounting principles." It is now compulsory for all forms of business enterprises to maintain a book of accounts. Accounts information helps in supplying the needed information for management and day to day running of the organization.

The General Ledger help in maintain the credit and debit status of a business. The importance of accounts and general ledger data has made it imperative for a more efficient method of recording and managing account day to be implemented. Thus therefore, an automated system would bring about the needed efficiency needed for office accounting system. An automated general ledger system that takes into consideration the principles of accounting and its operation in business organization would therefore of immense benefits in the entire business management process.

### **1.2 PROJECT OBJECTIVE**

The objective of this project is to develop an application program using a modern Object Oriented programming tool to enhance the Computation and recording of General Ledger Information.. The bottlenecks that are in the manual handling of general ledger are expected to be addressed and solved in this system.

### **1.3 METHODOLOGY**

The approach of this project was based on the following strategies

- i. Studying the existing systems and procedures
- ii. Identifying areas requiring improvement and modification for computation and generation of reports.
- iii. Develop an underlying system to timely management information system requirement in various areas.

- iv. The new system is to achieve some set goals that have not been embarked upon before in the institute under the present manual system.

#### 1.4 PROBLEM DEFINITION

General Ledger being an important subset of the entire accounting procedures have been bedeviled by several problems resulting from the improper storage and management of records of accounts information. These problems results from handling accounting records manually. Manual record handling is prone to several errors, which in turn makes the effective management of account records impossible.

#### 1.5 FEASIBILITY STUDY

This is a preliminary stage that determines whether or not the proposed project is desirable. Therefore, it involves the study of the existing system in detail in order to provide the management with information about the proposed computerized system.

The feasibility study covers three major areas:

- (a) Designing the computerized system
- (b) Testing and implementing the computerized system

To test project feasibility, the following are to be considered:

- i. **Operational Feasibility:** This relates to the workability of the proposed information system when developed and installed.
- ii. **Technical Feasibility:** This test, seeks to clarify if the proposed project can be done with current equipment, existing software technology and available personnel.
- iii. **Economic Feasibility:** The test for financial feasibility is undertaken to access cost of implementing a proposed project, that is, the benefit from implementing the project.

#### 1.5 FACTS FINDING TECHNIQUE

Three basic methods were employed to collect data:

- i. **Interview.** It is a fact-finding tool used for collecting information from individual. It involves holding discussion with people that are involved in the day to day operation of the system in order to ascertain their suggestions and comments and also the relevance of any

particular report or statement required from the department. Interview is one of the most satisfactory ways of obtaining information about objectives, constraint, allocation of duties, problems and failures in the existing system.

- ii. **Observation.** This involves watching an operation for a period, to see what happens by oneself.
- iii. **Special Purpose Records.** This involves, the examination of records that contains specific records in an organization, which may be files, journals, seminar papers or even textbooks.

## 1.6 SCOPE AND LIMITATION OF STUDY

This study concentrates on the General Ledger Operations of a business organization. It does not consider other aspects of accounting such as Cash Book Analysis, Assets Management, Accounts Payable and Receivable and Investments.

### 1.6 Definition of Terms

#### ***Account***

An individual record of specific things a business owns (assets) and owes (liabilities), as well as ownership (owner's equity).

#### ***Accounting***

The art of organizing, maintaining, recording, and analyzing financial information so users of that information can make better decisions.

#### ***Accounting Equation***

The relationship between the assets, liabilities, and capital of a business organization.  
(Assets=Liabilities+Owner's Equity)

#### ***Accounting Period***

The period of time, no more than one year, covered by the financial statements.  
You can have one year open at a given time.

### ***Account Payable***

A current liability for which an oral promise to pay, made to the creditor, serves as evidence of the debt.

### ***Account Receivable***

A current asset for which an oral promise to pay, made by the customer, serves as evidence of the debt.

### ***Accrual Basis***

An accounting system that recognizes the receipt of cash when it is earned rather than when it is actually received and records an expense when it is actually incurred rather than when the cash is disbursed.

### ***Adjusting Entries***

Journal entries recorded to change the appropriate balances in various accounts for a specific accounting period.

### ***Asset***

Anything owned and has monetary value.

### ***Bad Debt***

An expense a business recognizes as a result of a customer's failure to pay an obligation usually arising as a direct result of a prior credit sale.

### ***Balance***

The monetary value or worth of a specific ledger account. An individual account may be said to have a debit balance, a credit balance, or no (zero) balance.

### ***Balance Sheet***

A financial statement showing the financial position of a business at a particular moment in time. A detailed presentation of the assets, liabilities and owner's equity.



### ***Bank Reconciliation***

The process by which an account's balance as shown on the bank statement is brought into agreement with the balance shown on the depositor's records.

### ***Bank Statement***

A record sent by the bank, usually on a monthly basis, to indicate the bank's record of the activities within an individual account. Activities include, deposits, paid checks, various bank charges, collections made by the bank to the customer's account, and payments authorized from the customer's account.

### ***Book Value***

The original cost of a fixed asset, less the accumulated depreciation.

### ***Business Transaction***

Any business activity affecting what a business owns or owes, as well as the ownership of the business.

Capital The ownership of the assets of a business by the proprietors.

### ***Capital Expenditure***

A significant amount of money spent to increase the value or useful life of a fixed asset.

Cash The most liquid asset a business owns. These consist of coins, bills, money orders, checks, certificates of deposit or treasury bills.

### ***Cash Basis***

An accounting system that recognizes the revenue when the cash is received and the expense when the cash is spent. The system does not match the expense with the related revenue produced during the same accounting period. For example, you would record the revenue from a sale to a customer only when the customer pays you.

### ***Cash Discount***

A reduction in price offered by a seller to a buyer as an incentive to pay before the buyer is actually required to do so. See also, Discounts.

### ***Chart of Accounts***

A ledger's table of contents. A listing of the account numbers and account names in the ledger. The chart of accounts is traditionally set up in the order of the accounting equation: assets followed by liabilities, permanent capital and temporary capital accounts.

### ***Cheque***

An order to the bank on which it is written to pay a specific sum of money to a designated party.

## **CHAPTER TWO: LITERATURE RFEVIEW**

### **2.1 HISTORICAL STUDY – GENERAL BOOKKEEPING PRINCIPLES**

Accounting and Bookkeeping is the process of identifying, measuring, recording, and communicating economic information about an organization or other entity, in order to permit informed judgments by users of the information. Bookkeeping encompasses the record-keeping aspect of accounting and therefore provides much of the data to which accounting principles are applied in the preparation of financial statements and other financial information.

Personal record keeping often uses a simple single-entry system, in which amounts are recorded in column form. Such entries include the date of the transaction, its nature, and the amount of money involved. Record keeping of organizations, however, is based on a double-entry system, whereby each transaction is recorded on the basis of its dual impact on the organization's financial position or operating results or both. Information relating to the financial position of an enterprise is presented in a balance sheet, while disclosures about operating results are displayed in an income statement. Data relating to an organization's liquidity and changes in its financial structure are shown in a statement of changes in financial position. Such financial statements are prepared to provide information about past performance, which in turn becomes a basis for readers to try to project what might happen in the future.

#### **2.1.1. HISTORY**

Bookkeeping and record-keeping methods, created in response to the development of trade and commerce, are preserved from ancient and medieval sources. Double-entry bookkeeping began in the commercial city-states of medieval Italy and was well developed by the time of the earliest preserved double-entry books, from 1340 in Genoa.

The first published accounting work was written in 1494 by the Venetian monk Luca Pacioli. Although it disseminated rather than created knowledge about double-entry bookkeeping, Pacioli's work summarized principles that have remained essentially unchanged. Additional accounting works were published during the 16th century in Italian, German, Dutch, French, and English, and these works included early formulations of the concepts of assets, liabilities, and income.

The Industrial Revolution created a need for accounting techniques that were adequate to handle mechanization, factory-manufacturing operations, and the mass production of goods and services. With the emergence in the mid-19th century of large, publicly held business corporations, owned by absentee stockholders and administered by professional managers, the role of accounting was further redefined.

Bookkeeping, which is a vital part of all accounting systems, was in the mid-20th century increasingly carried out by machines. The widespread use of computers broadened the scope of bookkeeping, and the term *data processing* now frequently encompasses bookkeeping.

## 2.2 ACCOUNTING INFORMATION

Accounting information can be classified into two categories: financial accounting or public information and managerial accounting or private information. Financial accounting includes information disseminated to parties that are not part of the enterprise proper—stockholders, creditors, customers, suppliers, regulatory commissions, financial analysts, and trade associations—although the information is also of interest to the company's officers and managers. Such information relates to the financial position, liquidity (that is, ability to convert to cash), and profitability of an enterprise.

Managerial accounting deals with cost-profit-volume relationships, efficiency and productivity, planning and control, pricing decisions, capital budgeting, and similar matters. This information is not generally disseminated outside the company. Whereas the general-purpose financial statements of financial accounting are assumed to meet basic information needs of most external users, managerial accounting provides a wide variety of specialized reports for division managers, department heads, project directors, section supervisors, and other managers.

2.2.1. *Specialized Accounting*: of the various specialized areas of accounting that exist, the three most important are auditing, income taxation, and nonbusiness organizations. Auditing is the examination, by an independent accountant, of the financial data, accounting records, business documents, and other pertinent documents of an organization in order to attest to the accuracy of its financial statements. Businesses and not-for-profit organizations in the United States engage certified public accountants (CPAs) to perform audit examinations. Large private and public enterprises sometimes also maintain an internal audit staff to conduct auditlike examinations,

including some that are more concerned with operating efficiency and managerial effectiveness than with the accuracy of the accounting data.

The second specialized area of accounting is income taxation. Preparing an income-tax form entails collecting information and presenting data in a coherent manner; therefore, both individuals and businesses frequently hire accountants to determine their taxes. Tax rules, however, are not identical with accounting theory and practices. Tax regulations are based on laws that are enacted by legislative bodies, interpreted by the courts, and enforced by designated administrative bodies. Much of the information required in figuring taxes, however, is also needed in accounting, and many techniques of computing are common to both areas.

A third area of specialization is accounting for nonbusiness organizations, such as universities, hospitals, churches, trade and professional associations, and government agencies. These organizations differ from business enterprises in that they receive resources on some nonreciprocating basis (that is, without paying for such resources), they do not have a profit orientation, and they have no defined ownership interests as such. As a result, these organizations call for differences in record keeping, in accounting measurements, and in the format of their financial statements.

**2.2.2 Financial Reporting** Traditionally, the function of financial reporting was to provide proprietors with information about the companies that they owned and operated. Once the delegation of managerial responsibilities to hired personnel became a common practice, financial reporting began to focus on stewardship, that is, on the managers' accountability to the owners. Its purpose then was to document how effectively the owners' assets were managed, in terms of both capital preservation and profit generation.

After businesses were commonly organized as corporations, the appearance of large multinational corporations and the widespread employment of professional managers by absentee owners brought about a change in the focus of financial reporting. Although the stewardship orientation has not become obsolete, financial reporting in the mid-20th century is somewhat more geared toward the needs of investors. Because both individual and institutional investors view ownership of corporate stock as only one of various investment alternatives, they seek much more future-oriented information than was supplied under the traditional stewardship concept. As investors relied more on the potential of financial statements to predict the results of investment and

disinvestment decisions, accounting became more sensitive to their needs. One important result was an expansion of the information supplied in financial statements.

The proliferation of footnotes to financial statements is a particularly visible example. Such footnotes disclose information that is not already included in the body of the financial statement. One footnote usually identifies the accounting methods adopted when acceptable alternative methods also exist, or when the unique nature of the company's business justifies an otherwise unconventional approach.

Footnotes also disclose information about lease commitments, contingent liabilities, pension plans, stock options, and foreign currency translation, as well as details about long-term debt (such as interest rates and maturity dates). A company having a widely distributed ownership usually includes among its footnotes the income it earned in each quarter, quarterly stock market prices of its outstanding shares of common stock, and information about the relative sales and profit contribution of its different industry segments.

### 2.3 ACCOUNTING PRINCIPLES

Accounting as it exists today may be viewed as a system of assumptions, doctrines, tenets, and conventions, all encompassed by the phrase "generally accepted accounting principles." Many of these principles developed gradually, as did much of common law; only the accounting developments of recent decades are prescribed in statutory law. Following are several fundamental accounting concepts.

The *entity concept* states that the item or activity (entity) that is to receive an accounting must be clearly defined, and that the relationship assumed to exist between the entity and external parties must be clearly understood.

The *going-concern* assumption states that it is expected that the entity will continue to operate indefinitely.

The *historical-cost principle* requires that economic resources be recorded in terms of the amounts of money exchanged; when a transaction occurs, the exchange price is by its nature a measure of the value of the economic resources that are exchanged.

The *realization concept* states that accounting takes place only for those economic events to which the entity is a party. This principle therefore rules out recognizing a gain based on the appreciated market value of a still-owned asset.

The *matching principle* states that income is calculated by matching a period's revenues with the expenses incurred in order to bring about that revenue.

The *accrual principle* defines revenues and expenses as the inflow and outflow of all assets—as distinct from the flow only of cash assets—in the course of operating the enterprise.

The *consistency criterion* states that the accounting procedures used at a given time should conform with the procedures previously used for that activity. Such consistency allows data of different periods to be compared.

The *disclosure principle* requires that financial statements present the most useful amount of relevant information—namely, all information that is necessary in order not to be misleading.

The *substance-over-form* standard emphasizes the economic substance of events even though their legal form may suggest a different result. An example is the practice of consolidating the financial statements of one company with those of another in which it has more than a 50 percent ownership interest.

The *conservatism doctrine* states that when exposure to uncertainty and risk is significant, accounting measurement and disclosure should take a cautious and prudent stance until evidence shows sufficient lessening of the uncertainty and risk.

a. *The Balance Sheet* of the two traditional types of financial statements, the balance sheet relates to an entity's position, and the income statement relates to its activity. The balance sheet provides information about an organization's assets, liabilities, and owners' equity as of a particular date (such as the last day of the accounting or fiscal period). The format of the balance sheet reflects the basic accounting equation: Assets equal equities. Assets are economic resources that provide potential future service to the organization. Equities consist of the organization's liabilities together with the equity interest of its owners. (For example, a certain house is an asset worth N70,000; its unpaid mortgage is a liability of N45,000, and the equity of its owners is N25,000.)

Assets are categorized as current or long-lived. Current assets are usually those that management could reasonably be expected to convert into cash within one year; they include cash, receivables, merchandise inventory, and short-term investments in stocks and bonds. Long-lived assets encompass the physical plant—notably land, buildings, machinery, motor vehicles, computers, furniture, and fixtures. Long-lived assets also include real estate being held for speculation and intangibles such as patents and trademarks.

Liabilities are obligations that the organization must remit to other parties, such as creditors and employees. Current liabilities usually are amounts that are expected to be paid within one year, including salaries and wages, taxes, short-term loans, and money owed to suppliers of goods and services. Noncurrent liabilities are usually debts that will come due beyond one year—such as bonds, mortgages, and long-term loans. Whereas liabilities are the claims of outside parties on the assets of the organization, the owners' equity is the investment interest of the owners in the organization's assets. When an enterprise is operated as a sole proprietorship or as a partnership, the balance sheet may disclose the amount of each owner's equity. When the organization is a corporation, the balance sheet shows the equity of the owners—that is, the stockholders—as consisting of two elements: (1) the amount originally invested by the stockholders; and (2) the corporation's cumulative reinvested income, or retained earnings (that is, income not distributed to stockholders as dividends), in which the stockholders have equity.

*b. The Income Statement.* The traditional activity-oriented financial statement issued by business enterprises is the income statement. Prepared for a well-defined time interval, such as three months or one year, this statement summarizes the enterprise's revenues, expenses, gains, and losses. Revenues are transactions that represent the inflow of assets as a result of operations—that is, assets received from selling goods and rendering services. Expenses are transactions involving the outflow of assets in order to generate revenue, such as wages, rent, interest, and taxes.

A revenue transaction is recorded during the fiscal period in which it occurs. An expense appears in the income statement of the period in which revenues presumably resulted from the particular expense. To illustrate, wages paid by a merchandising or service company are recognized as an immediate expense because they are presumed to generate revenue during the same period in which they occurred. If, however, the wages are paid to process merchandise that will not be sold until a later fiscal period, they would not be considered an immediate expense. Instead, the cost of



these wages will be treated as part of the cost of the resulting inventory asset; the effect of this cost on income is thus deferred until the asset is sold and revenue is realized.

In addition to disclosing revenues and expenses (the principal components of income), the income statement also lists gains and losses from other kinds of transactions, such as the sale of plant assets (for example, a factory building) or the early repayment of long-term debt. Extraordinary—that is, unusual and infrequent—developments are also specifically disclosed.

*c. Other Financial Statements:* The income statement excludes the amount of assets withdrawn by the owners; in a corporation such withdrawn assets are called dividends. A separate activity-oriented statement, the statement of retained earnings, discloses income and redistribution to owners.

A third important activity-oriented financial statement is the statement of cash flows. This statement provides information not otherwise available in either an income statement or a balance sheet; it presents the sources and the uses of the enterprise's funds by operating activities, investing activities, and financing activities. The statement identifies the cash generated or used by operations; the cash exchanged to buy and sell plant and equipment; the cash proceeds from stock issuances and long-term borrowings; and the cash used to pay dividends, to purchase the company's outstanding shares of its own stock, and to pay off debts.

*d. Bookkeeping and Accounting. Cycle* Modern accounting entails a seven-step accounting cycle. The first three steps fall under the bookkeeping function—that is, the systematic compiling and recording of financial transactions. Business documents provide the bookkeeping input; such documents include invoices, payroll time cards, bank checks, and receiving reports. Special journals (daily logs) are used to record recurring transactions; these include a sales journal, a purchases journal, a cash-receipts journal, and a cash-disbursements journal. Transactions that cannot be accommodated by a special journal are recorded in a general journal.

#### **d1. Step One**

Recording a transaction in a journal marks the starting point for the double-entry bookkeeping system. In this system the financial structure of an organization is analyzed as consisting of many interrelated aspects, each of which is called an account (for example, the "wages payable" account). Every transaction is identified in two aspects or dimensions, referred to as its debit (or

left side) and credit (or right side) aspects, and each of these two aspects has its own effect on the financial structure. Depending on their nature, certain accounts are increased with debits and decreased with credits; other accounts are increased with credits and decreased with debits. For example, the purchase of merchandise for cash increases the merchandise account (a debit) and decreases the cash account (a credit). If merchandise is purchased on the promise of future payment, a liability would be created, and the journal entry would record an increase in the merchandise account (a debit) and an increase in the liability account (a credit). Recognition of wages earned by employees entails recording an increase in the wage-expense account (a debit) and an increase in the liability account (a credit). The subsequent payment of the wages would be a decrease in the cash account (a credit) and a decrease in the liability account (a debit).

#### **d2. Step Two**

In the next step in the accounting cycle, the amounts that appear in the various journals are transferred to the organization's general ledger—a procedure called posting. (A ledger is a book having one page for each account in the organization's financial structure. The page for each account shows its debits on the left side and its credits on the right side, so that the balance—that is, the net credit or debit—of each account can be determined.)

In addition to the general ledger, a subsidiary ledger is used to provide information in greater detail about the accounts in the general ledger. For example, the general ledger contains one account showing the entire amount owed to the enterprise by all its customers; the subsidiary ledger breaks this amount down on a customer-by-customer basis, with a separate subsidiary account for each customer. Subsidiary accounts may also be kept for the wages paid to each employee, for each building or machine owned by the company, and for amounts owed to each of the enterprise's creditors.

#### **d3 Step Three**

Posting data to the ledgers is followed by listing the balances of all the accounts and calculating whether the sum of all the debit balances agrees with the sum of all the credit balances (because every transaction has been listed once as a debit and once as a credit). This determination is called a trial balance. This procedure and those that follow it take place at the end of the fiscal period.

Once the trial balance has been successfully prepared, the bookkeeping portion of the accounting cycle is concluded.

#### **d4 Step Four**

Once bookkeeping procedures have been completed, the accountant prepares certain adjustments to recognize events that, although they did not occur in conventional form, are in substance already completed transactions. The following are the most common circumstances that require adjustments: accrued revenue (for example, interest earned but not yet received); accrued expense (wage cost incurred but not yet paid); unearned revenue (earning subscription revenue that had been collected in advance); prepaid expense (expiration of a prepaid insurance premium); depreciation (recognizing the cost of a machine as expense spread over its useful economic life); inventory (recording the cost of goods sold on the basis of a period's purchases and the change between beginning and ending inventory balances); and receivables (recognizing bad-debt expenses on the basis of expected uncollected amounts).

#### **d5 Steps Five and Six**

Once the adjustments are calculated, the accountant prepares an adjusted trial balance—one that combines the original trial balance with the effects of the adjustments (step five). With the balances in all the accounts thus updated, financial statements are then prepared (step six). The balances in the accounts are the data that make up the organization's financial statements.

d6 Step Seven The final step is to close noncumulative accounts. This procedure involves a series of bookkeeping debits and credits to transfer sums from income-statement accounts into owners' equity accounts. Such transfers reduce to zero the balances of noncumulative accounts so that these accounts can receive new debit and credit amounts that relate to the activity of the next business period.

## **2.4 BENEFITS OF AN AUTOMATED LEDGER SYSTEM**

In bringing about the much needed efficiency in the administration of Ledger information. Computers and other Office Systems need to be applied as a replacement for the manual system.

An automated general ledger systems would offer the following advantages;

- Easy computation of accounting information.
- Finding specific records with little effort by the use of search criteria

- Access to stored information in many different ways
- Generation of multiple reports from databases
- Avoidance of unnecessary duplication of records
- More efficient management of stored information
- Protection of database from unauthorized access
- Regular back-up of data files

For these advantage to be achieved, an effective data processing system needs to put in place. this should be complemented with an organized office system.

## **2.5 UNDERLINING CONCEPTS OF AUTOMATED SYSTEMS**

Data processing is the analysis and organization of data by the repeated use of one or more computer programs. Data processing is used extensively in business, engineering and science and to an increasing extent in nearly all areas of human activities. Businesses use data processing for such tasks as payroll preparation, accounting, record keeping, inventory control and sales analysis. Engineers and scientists use data processing for a wide variety of applications, including the processing of seismic data for oil and mineral exploration, the analysis of new product designs, the processing of satellite imagery, and the analysis of data from scientific experiments.

Data processing is divided into two kinds of processing:

- Database processing
- Transaction processing

### **Database and Transaction Processing**

A database is a collection of common records that can be searched, and modified such as bank account records, school transcript and income tax data. In database processing a computerized database is used as the central source of reference data for the computations.

Transaction Processing refers to interaction between two computers in which one computer initiates a transaction and another computer provides the first with the data required for that function.

Most modern data processing uses one or more databases at one or more central sites. Transaction Processing is used to access and update the databases when users need to immediately view or add information; other data processing programs are used at regular intervals to provide summary reports

of activity and database status. Examples of systems that involve all of these functions are automated teller machines, credit sales terminals and airline reservation systems.

### **Information Storage and Retrieval**

The final step in data processing cycle is the retrieval of stored information at a later time. It is a term used to describe the organization, storage, location and retrieval of encoded information in computer systems.

Important factors in storing and retrieving information are the:

- Type of media
- Storage device used to store information
- The media's storage capacity
- The speed of access and information transfer to and from the storage media
- The number of times new information can be written to the media

This is usually done to access records contained in a database, to apply new data-processing functions to the data, or - in event that some part of the data has been lost – to recreate portions of a database. Examples of data retrieval in the data-processing cycle include the analysis of stored sales receipts to reveal new customer spending patterns and the application of new processing techniques to seismic data to locate oil or mineral fields that were previously overlooked.

## **CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN**

### **3.1 PLANNING FOR A COMPUTER BASED LEDGER SYSTEM**

The method selected for the new system is determined by the present set of document used, which is fairly adequate to be modified with the design of new formats to form the basis for the computerized information system. The files and programs are kept as simple as possible to be users friendly in relation to computerization of a ledger system.

The initial step toward system design is the identification of system requirements and the formulation of design alternative. The requirements are those factors or details that have to be incorporated into the proposed system to provide desired output. The following are borne in mind during this system design:

- Identifying areas that require improvements and/or modifications for computerization
- Development of an underlying system to achieve the computerization requirements.
- Ensuring that the proposed system is easy to operate, understood by all user staff, and which requires minimum changes in the existing system.

In designing the system, the following strategies are adopted:

#### **3.1.1 Design Strategies For Implementing Computer Based Ledger System.**

- Identification of current system requirements
- Selection and Organization of the data that needs to be stored
- Planning for system Efficiency and Reliability
- Economic Cost comparison: Determination of the cost of implementing the proposed system

### **3.1.2 Planning System Requirement**

- Determination of the information or data to be keep track off.
- Organization strategy of the required information
- Determination of the capacity of current hardware and software that the proposed system would use.
- Determination of the user of the proposed system and the capability of such user.
- Projection of future database needs.

### **3.1.3 Potential Hurdles**

- Apathy of users to computerized systems, leading to the exhibition of system under-utilization.
- Over dependency leading to the assumptions that the computerized system is fail-prove.
- Unauthorized access to stored data and information.
- Data lost due to user lack of maintenance culture
- System malfunctioning.
- User adaptation to old system.

### **3.1.4 Solving Potential Problems**

- Adequate training of operators, users and all personnel concerned with the use of the new system.
- Periodic/Routine backup of stored information.
- Access control and user verification methods should be built into the software.
- Management Control: Management should always summarize information, observe trends and performance of variances.

### 3.2 FEATURES OF USED PROGRAMMING LANGUAGE (VISUAL BASIC 6.0)

The proposed system would be developed with modern Object Oriented Programming Tools (Visual Basic) and would very much help in enhancing any standard operations.

The Visual Basic has a number of features, which in turn are incorporated into the proposed system, this feature includes:

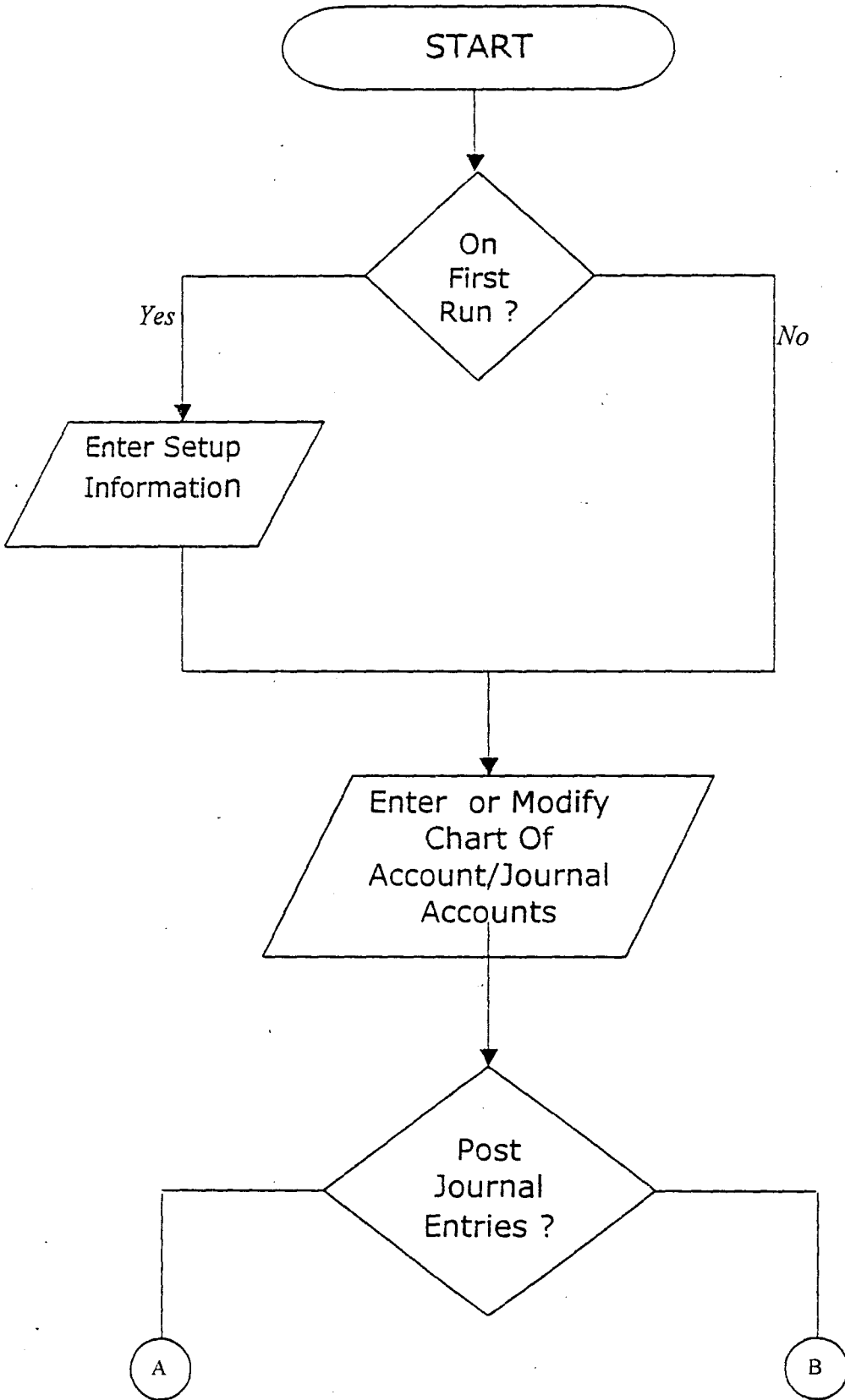
#### FEATURES

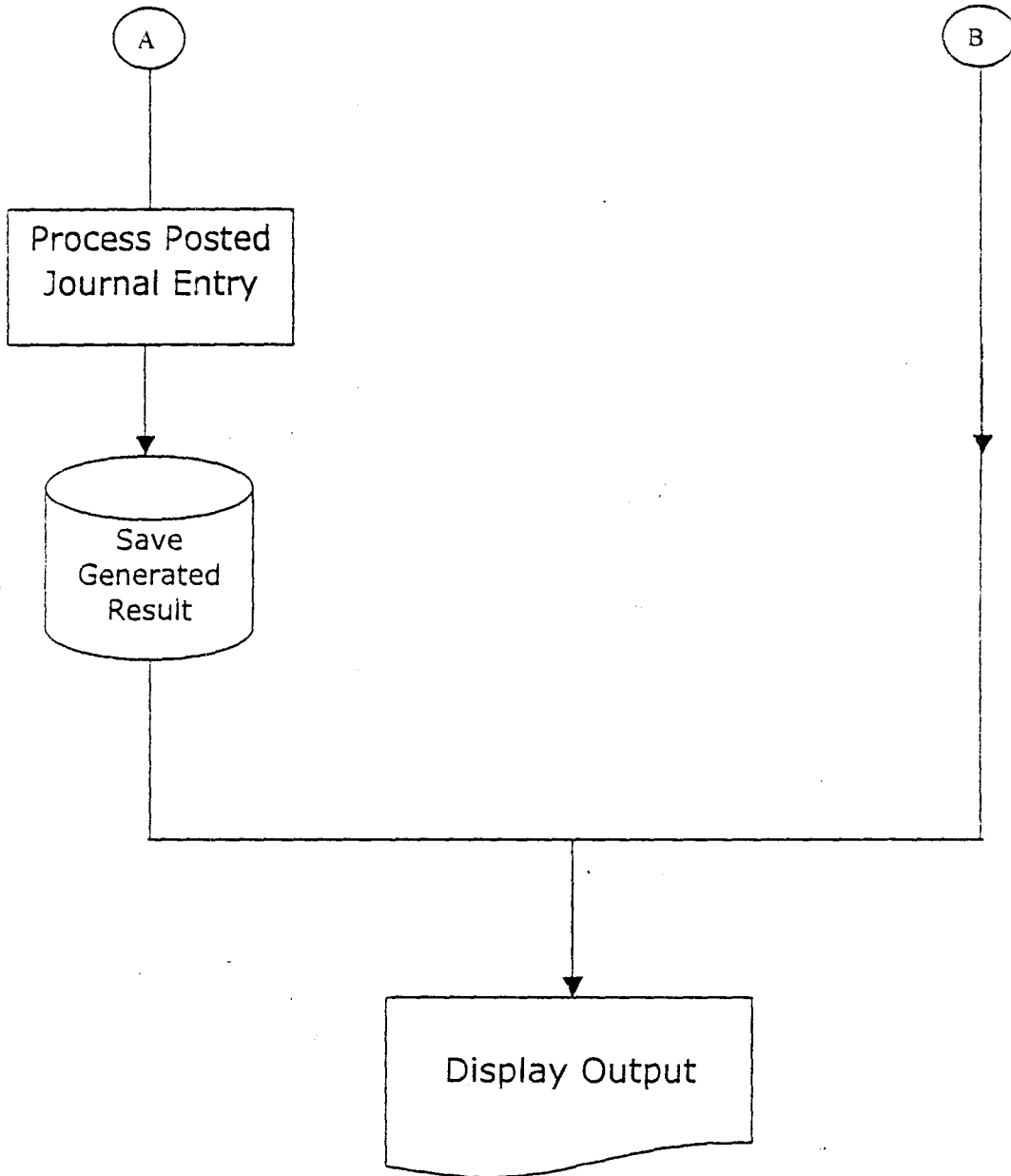
- **Full Windows Compatibility:** the software is designed to run in the Ms Windows Operating and any compatible environment, which is the standard for any modern software.
- **Mouse Support :** apart from keyboard invocations of commands, the software we develop supports the use of the mouse, hence, it is easier to navigate through the software environment and tasks can be executed at the click of a button.
- **Menu and Event Driven:** the software incorporate Pull Down menus, windows compatible dialogue boxes, prompts and command buttons. This has the added advantage of making the software more user-friendly and interactive.
- **Connectivity with other Windows based application:** the software has the ability for Dynamic Data Exchange (DDE) and Dynamic Link Library (DLL) for Shared files, with other Ms Windows based software. As such data can be exported or imported to and from it (i.e. communication with other window based software is totally effective).
- **Improved Multiple Database Structure:** the database structure of any new software developed by us uses the Microsoft Open Database Connectivity (ODBC) style that supports the creation of multiple tables in a single database. This allows for better handling of large volume of related data items while still compact.
- **Security:** the software is designed such that only valid and authorized users can only gain access; thereby protecting the integrity of your data and the source code of your program being tampered with, which is one of the problems you faced with uncompileable program. Also the databases in the software could encrypted such that they can only be manipulated by the software alone.



- ***Client-Server Network System (Optional):*** the software is designed to work on Network environment of any architecture. It makes database access in the multi-user environment totally efficient, protecting data Integrity and Redundancy.
- ***Output Control:*** the software controls the output to Screen, Printer or directly to a file in the format of any Ms Windows base standard software. (E.g. Access, Excel, Ms Word)
- ***Practical WYSIWYG (What-You-See-Is-What-You-Get) Report Display:*** the display screen shows you exactly what your final report will look like when you print
- ***Multiple Size Page Views:*** View as many report pages at Actual, Double or Half size.
- ***Pictorial Data Representation:*** data charting in our software gives view enhancement.
- ***Multi-Media Effects:*** Visual and Audio multi-media are available optionally, also they use all your current Operating System Settings like Sound, Screen Color, Date & Time e.t.c.
- ***Utility Features:*** pop-up on screen Calculator & Calendar.
- ***Communication (Internet):*** automatic E-mailing of Report/Data available as an additional report route.
- ***Online-Help:*** Help is available for users on virtually every command of our software. Appropriate error trapping messages are included in our systems.

3.3 SYSTEM FLOWCHART





### 3.4 INPUT SPECIFICATION

The input requirements of a computerized ledger system are analyzed and specified as below:

**Table Name: GLCHART**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
Gcode	Text	8
Glname	Text	255
Gtype	Number	Single
Gcontrol	Text	8
Gcustno	Text	255
Glstatus	Yes/No	
Glevel	Text	255

***Index:*** The Table is index upon gcode with a Unique attribute

**Table Name: GLLEDGER**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
GLCODE	Text	8
Glp01	Currency	
Glp02	Currency	
Glp03	Currency	
Glp04	Currency	
Glp05	Currency	
Glp06	Currency	
Glp07	Currency	
Glp08	Currency	
Glp09	Currency	
Glp10	Currency	
Glp13	Currency	
Glyear	Number	Long Integer

**Table Name: GLJNLBUP**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
AcctYear	Num	Long Integer
Aperiod	Num	Long integer
AcctInType	Text	3
AcctSerial	Number	Long Integer
AcctCode	Text	8
Ref. No	Text	50
Desc	Text	100
DR	Currency	
CR	Currency	
Source	Text	50
Date	Date/Time	Medium Date
Posted	Yes/No	
User	Text	30
Name	Text	40

**Table Name: AcctLevel**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
Code	Text	4
Desc	Text	100

**Index:** The Table is index on both fields (code and desc.)

**Table Name: glJournal**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
AcctYear	Num	Long Integer
Aperiod	Num	Long integer
AcctjnType	Text	3
AcctSerial	Number	Long Integer

AcctCode	Text	8
Ref. No	Text	50
Desc	Text	100
DR	Currency	
CR	Currency	
Source	Text	50
Date	Date/Time	Medium Date
Posted	Yes/No	
User	Text	30
Name	Text	40

**Table Name: AcctType**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
Code	Number	Single
Desc	Text	100

**Index:** The Table is index on both fields (code and desc.)

**Table Name: JnHist**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
AcctYear	Num	Long Integer
Aperiod	Num	Long integer
AcctjnType	Text	3
AcctSerial	Number	Long Integer
AcctCode	Text	8
Ref. No	Text	50
Desc	Text	100
DR	Currency	
CR	Currency	
Source	Text	50
Date	Date/Time	Medium Date

Posted	Yes/No	
User	Text	30
Name	Text	40

**Table Name: AcctType**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
Code	Text	100
Code	Text	3

**Table Name: GIBudget**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
GLCODE	Text	8
Glp01	Currency	
Glp02	Currency	
Glp03	Currency	
Glp04	Currency	
Glp05	Currency	
Glp06	Currency	
Glp07	Currency	
Glp08	Currency	
Glp09	Currency	
Glp10	Currency	
Glp13	Currency	
Glyear	Number	Long Integer

**Table Name: Users**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
Username	Text	100
Password	Text	100

AccessLevel	Number	Long Integer
-------------	--------	--------------

**Index:** The Table is indexed on the Username and Password fields.

### 3.5 OUTPUT SPECIFICATION

For the output of processing to be view (on Paper or Screen) they have to be first written to a file (Database Table) before they are sent to any output medium (Printer- for Hardcopy or Screen – for Softcopy). It is however pertinent to pre-empt and specify the output of the Computerized general ledger system.

It should be noted that most (in fact all) of this Table are generic, with the sole aim of ‘dumping’ Report data/information and are deleted afterwards. It is also note-worthy that a generated table could be used for/by many Reports.

Since the Tables are generated and deleted, they are given the same name “TRASHBIN”

The Output specifications for some Reports are:

**Table Name: TrashBin**

<i>Field Name</i>	<i>Field Type</i>	<i>Field Width</i>
CurBal	Currency	
Pbal	Currency	
GlCode	Text	255
GlYear	Number	Single
GlName	Text	255

**Index:** The Table is indexed on the Username and Password fields.

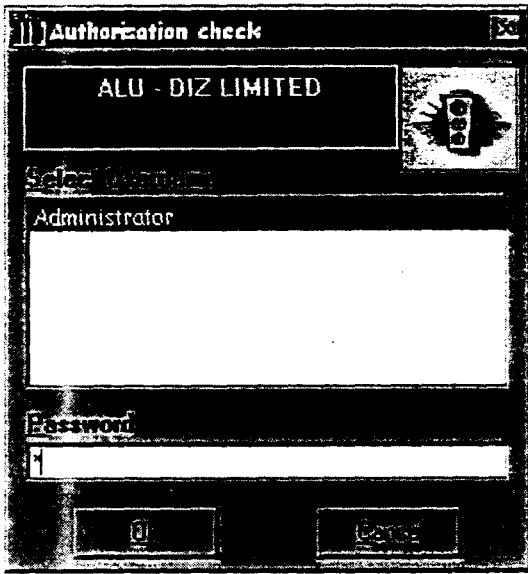
#### 3.5.1 INPUT DESIGN

The Design of Screens (FORMS as called in Visual Basic) is important in any system development process, because it is through this Forms (interface) that the User actually communicates with the program, thus, the efficiency or robustness of a program is firstly determined by the User interface (Forms). It is the user friendliness of the Forms in a program that determines whether is Software is good or not. With this taken into consideration, the following input designs are used.

**Form Name: FRMPASSA.FRM**

**Screen Purpose:** This Form allows the user to enter his/her name and password. The software verifies that the user is dully registered and if the user is authorized, the software loads the main form.





**Form Name: FRMMAINLEDGER.FRM**

**Screen Purpose:** This is the Main menu Form that contains the different activities the system can do. Different operational options are available on the main menu form for users to choose. The main menu contains the following Main options and sub-menus:



MHlegder 2000<sup>+</sup> Main Screen is a simple interface that contains typically:

Menu bar, Toolbar and a Status bar.

### MENU BAR

The Menu Bar is the top most bar on the screen, it consist of the following main menu items:

- FILE

- *Backup/Restore*: This is a command option for Data file backup and restore. The Software utilize the WinZip Zip utility which is shipped with MHlegder 2000<sup>+</sup>
- *Set Default Printer*: This sub-menu option sets an installed Printer as the default printer to be used for reports to be displayed on screen or paper. The type of printer chosen would sometimes determines the setting of the paper to be used in printing. All reports are designed using DeskJet A4 paper settings
- *Exit Package*: This option closed the software session.

- MAINTAIN

- *Chart of Account*: The Chart of Account menu option is the command option that invokes the Chart of Account setup form for either new account registration, recent account modification or account deletion.

- *Journal Entry*: This is the option that invokes the journal entry main form.
- *Budgets*: The Budget sub menu option invokes Budget control module.
- *Account Type*: The Account type setup routine is invoked using this menu option. However, as explained above, for obvious reasons, this menu option is visible but disabled, the menu option can be enabled by double clicking the main menu form.
- *Journal Type*: This option invokes the form for entry, editing, deleting and displaying Journal types.
- *Account Level*: The Account Level setup routine is invoked using this menu option. As explained above, for obvious reasons, this menu option is visible but disabled, the menu option can be enabled by double clicking the main menu form.
- *Delete All Files*: The Option deletes all records in the database. However, this operation is VERY dangerous and very much unnecessary unless unavoidably required.

- PROCESS

- *Post Journal*: This Option invokes a routine for Posting current Journal transactions.
- *Year End*: This option performs End Of Year operations

- REPORTS

The various reports are sub-menu to this main menu, the reports are:

- Budget List
- GL Activities
- Vote Book
- Audit Trail
- Income Statement
- Profit & Lost Account
- Trial Balance
- Balance Sheet

- TOOLS

- *Password Maintenance*: This option invokes the authorization routine, where the software users can be registered.
- *Calculator*: This option invokes a calculator utility.

- HELP

## TOOL BAR

Ledger Tool bar consist of 6 different icons, each for different purpose. The Toolbar contains frequently accessed procedure and routines, the tool bar consist of:



This is the Tool bar item that invokes the Chart of Account form.



Displays the Journal Entry Screen



Displays the Budget control routine form



Invokes the Journal Posting procedure



Performs the End-of-Year operation



Exit Package

## STATUS BAR

The Status Bar is the last bar and it is at the bottom of the screen. it consist such information like the:

- User name
- Current System date
- Current System Time
- Software Programmer Details

The information on the Status bar are displayed for the verification of the above information since they would be used as record 'stamps'.

## Ledger<sup>+</sup> Utility Operations

Ledger<sup>+</sup> has some operations that are independent of the accounting procedures but nevertheless important to the overall efficiency of the entire Software.

The utility operations includes:

- Backup/Restore

- Setting of Default Printer
- Password Maintenance
- Calculator

---

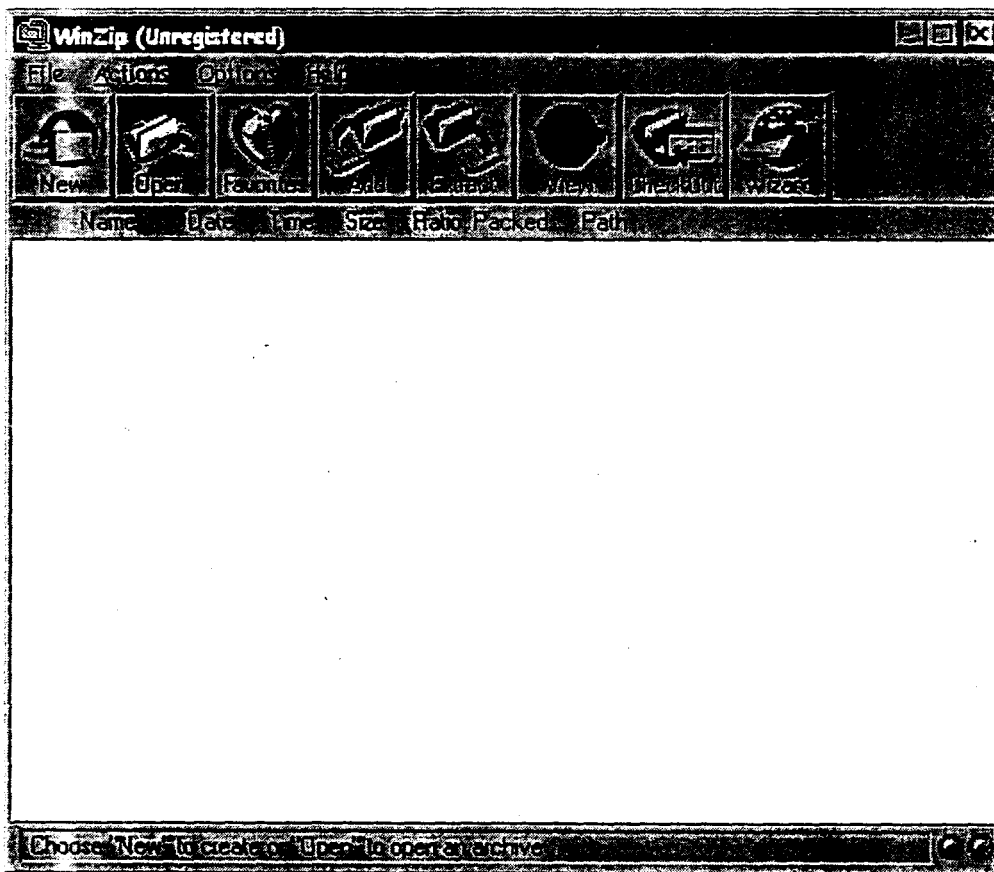
## BACKUP/RESTORE

---

This Utility routine allows the backing up of data files while compressing it the data file adequately to fit into a storage medium. The zipped data file can also be unzipped to Restore it for use by Ledger<sup>+</sup>.

The Zip utility WinZip which is shipped with Ledger<sup>+</sup> is used for this utility routine.

To Backup or Restore Click on the File menu and Backup/Restore sub-menu of Ledger 2000<sup>+</sup>. The interface below is displayed



### To Backup and Zip

1. Click on the New Icon on the Toolbar of the WinZip menu



**Journal Type**

Code: JV

Description: Journal Voucher

Buttons: New, Edit, Refresh, Delete, etc.

**Form Name: FRMCHART.FRM**

*Screen Purpose:* The Chart of Account defines all the various accounts that will be transacted. It is the Chart of Account setup that allows for the entering of both General and Detail accounts of various account types (Fixed Asset, Liability, Income, Expenditure e.t.c.).

**Chart of Account**

File: FRMCHART.FRM

Control/General Accounts:

Account Code: 103

Account Name: CURRENT ASSET & CURRENT LIABILITIES

Account Type: 2

Buttons: New, Edit, Refresh, Delete, etc.

Detail/Accounts:

Acc. Code	Acc. Name	Status
10301	Work-In-Progress	True
10302	Staff Advances	True
10303	Prepayment-Rent	True
10304	Julius Berger PLC	True
10305	Julius Berger Retention	True
10306	VAT Input A/C	True
10307	Bank+Cash	True
10308	NHF	True
10309	Staff Creditors	True
10310	Contract Advance	True

The Chart of Account setup screen is divided into two sections;

- The First section allows entry of General/Control accounts, where it is expected that the Account Code, Account Description and the Account Type should be supplied.

- The Second section allows the entry of Detail accounts that is controlled by the current General account. In the grid, it is expected that the Detail Account Code, Detail Account Description and the Status (Sets active or Inactive status of the account) be entered. A Detail account with a FALSE Status would not be transact-able until set TRUE, the default is FALSE.

A Detail Account can only have one Control Account, and its account type is the account type of the Controlling General account.

No two accounts, whether General or Detail must have the same Account Code.

On the Chart of Account screen is a menu 'File' and 'Find Account'.

The *File* menu allows the display of the Chart of Account on Screen or Printer.

The *Find Account* menu allows the 'quick' searching of accounts either using Account code or Account name referencing. Wild cards (\*, ?) can be used in the search string.

---

## SETTING DEFAULT PRINTER

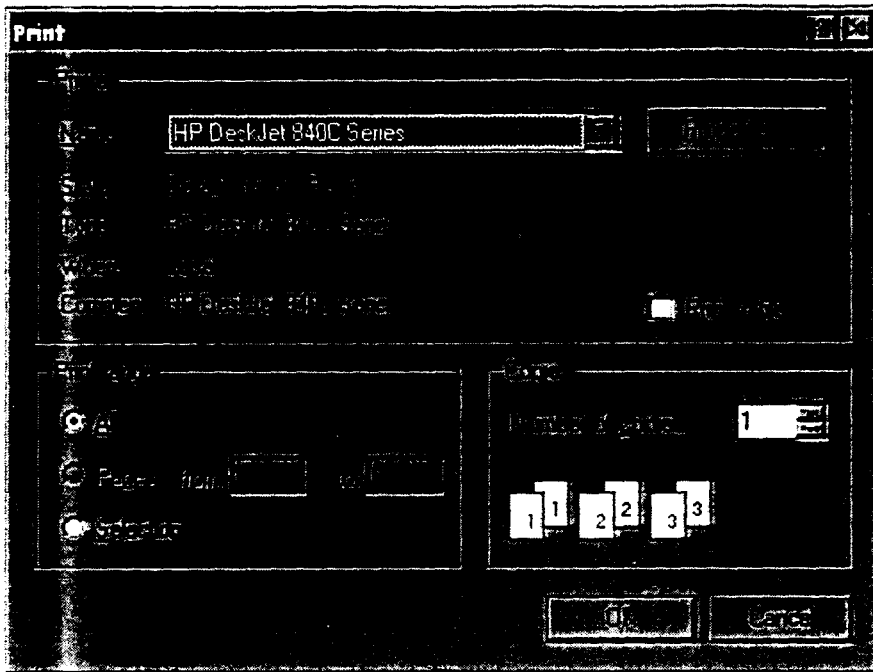
---

This routine sets the an installed printer as the default to be used in report display/printing, The type of printer chosen would sometimes determines the setting of the paper to be used in printing. All reports are designed using DeskJet A4 paper settings.

To set default printer:

- Click on File menu
- Click on set Default Printer





Note that the print dialog that appears depends on the standard print dialog of the installed printer or the operating systems print dialog box.

---

## PASSWORD MAINTENANCE

---

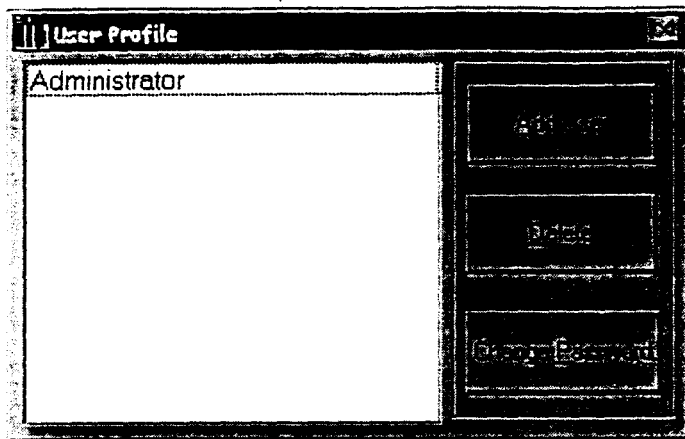
The System Administrator must register any user of Ledger<sup>+</sup> before the user can have access to the program. All registered users can be deleted except the Administrator.

The password maintenance routine allows for new user registration, recent user deletion and user password editing.

To invoke Password maintenance routine:

- Click on Tools
- Click on set Password Maintenance

**Form Name: FRMPASSP.FRM**

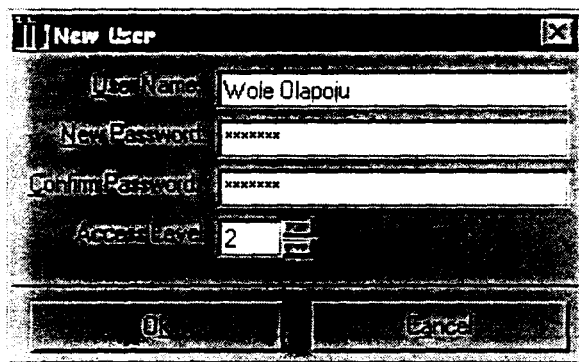


*Screen Purpose:* The User Profile screen appears where a new user can be added, Change Password or Delete.

**Form Name: FRM.PASSP.FRM**

### Adding New User

Click the Add User command button on the User Profile screen.



To add a new user profile the User name, the New Password, Confirm Password and the Access Level must be supplied.

The New Password and the Confirm Password must be the same for the registration information to be accepted. The Access Level is an index that defines the scope the software operation of the new user.

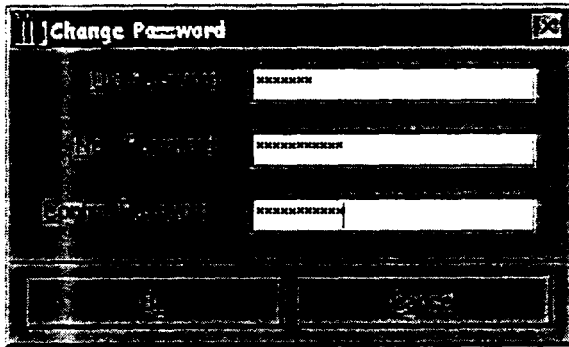
### Deleting User Profile

Select the User name to delete in the User Profile screen then click the Delete command button.

**Form Name: frmPassC.frm**

### Change Password

Select the User name to edit its Password in the User Profile screen clicking the Change Password button command.



To change an existing user profile the Old Password, the New Password, Confirm Password must be supplied.

The New Password and the Confirm Password must be the same for the registration information to be accepted. The Access Level is an index that defines the scope the software operation of the new user.

The Old Password must be the same as the password that requires changing.

The New Password and the Confirm Password must be the same for the editing information to be accepted.

---

## JOURNAL MANGEMENT

---

*Form name:* **FrmJournal.frm**

Screen Purpose:        Allows a user define select account code and enter transaction particulars.

**JOURNAL**

Account Year: 2001    Account Period: 1    Transaction Date: 06-Jan-2001

Journal Type: JV    Voucher Number: 5001

LINE	ACCOUNT CODE	DESCRIPTION	DR	CR
1	10201	JV001 Travel to New York	15,263,255.00	
2	10202	JV002 Chairman's Donation		15,000,000.00
3	10301	Cash Purchases	263,255.00	
4	10303	Cash Remittance		526,510.00
5	10303 - Prepayment-Rent			

10303 - Prepayment-Rent  
 10304 - Julius Berger PLC  
 10305 - Julius Berger Retention  
 10306 - VAT Input A/C  
 10307 - Bank+Cash  
 10308 - NHF  
 10309 - Staff Creditors  
 10401 - Called up Share Capital

Save    Edit    Close    Balance: 0

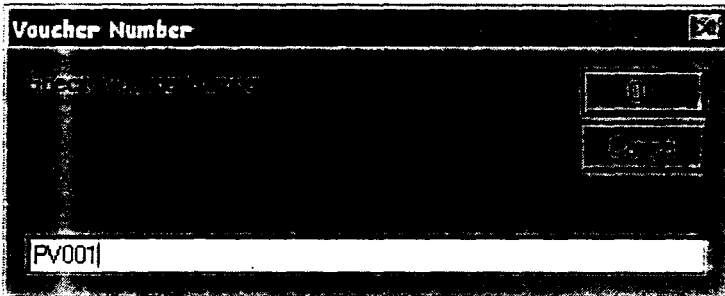
For a Ledger processing the Journal entry form provides the forum for entry all relevant information required in any standard Ledger operation. The Journal entry form requires the following information:

- *Account Year* – by default Ledger<sup>+</sup> uses the current system year
- *Account Period* – by default Ledger<sup>+</sup> uses the current system month representing them as numerals (i.e. 1 – Jan, 2 – Feb, 3 – Mar etc.)
- *Transaction Date* – by default Ledger<sup>+</sup> uses the current system date
- *Journal type* – this is selection of the type of journal from the various types defined in the Journal Type setup.
- *Voucher Number* – this is a unique number used in identifying Journal Vouchers. No two vouchers number must be the same.
- *Line Number* – this is an automatically generated number that itemizes the various transacted accounts in a Ledger.
- *Account Code* – this is the code of the account to be transacted in the Ledger.
- *Description* – this is the additional information about the transacted account in the Ledger.
- *DR* – this is the Debit amount of the transaction

- CR – this is the Credit amount of the transaction

**Form Name: FRMNEW.FRM**

*Screen Purpose:* Allows a user to create a new voucher. If the supplied Voucher number existing, an error prompt is displayed.

A screenshot of a software form titled "Voucher Number". The form has a dark background with white text and input fields. At the top left, the title "Voucher Number" is displayed. Below the title, there are several input fields. The most prominent one at the bottom is a wide text box containing the value "PV001". To the right of this text box, there are two smaller, stacked input fields, the top one containing "01" and the bottom one containing "000". The form also features a small "X" icon in the top right corner, likely for closing the window.

After the supplied Voucher number verification is successful (when non-existent) then the Journal type is requested, note if the Journal type is not specified then the grid to accept other information will not be enabled.

In the grid it is expected that the Account Code (which can be pick from the combo that contains both account code and description), transaction description and the Dr or Cr amount be supplied for every line of transaction. The line number is generated and cannot be supplied by the user; they are no course for alarm if the line number takes a wrong sequence.

While entering the transaction lines in the Ledger, Ledger<sup>+</sup> automatically calculated the balance and the result of such calculation decides whether the transaction could be saved or not. THE LEDGER TRANSACTION CAN ONLY BE SAVED IF THE BALANCE IS ZERO (0).

Editing of Current Ledger (i.e. Ledger transactions not yet Posted) is possible in the Journal Entry form be clicking the Edit command button. A request of the voucher number of the Ledger to edit is made, and if found it is displayed on the Journal entry form. NOTE THAT AFTER THE CALL FOR EDITTING ALWAYS SAVE EVEN WHEN MODIFICATION WAS NOT MADE.

It is note worthy to know that all line of transaction is 'stamped' with the user name of the user that login into the program.

**Form Name: FRMBUDGET.FRM**

*Screen Purpose:* The Budget control allows Ledger+ to monitor budgeted amount for all defined accounts, and compare such amounts with the actual amount expended on such account. The Budget control allows a one-view comparison of the Actual and Budgeted amount of an account. The budget control takes in budget per annum and splits them into the 12 periods in the year.

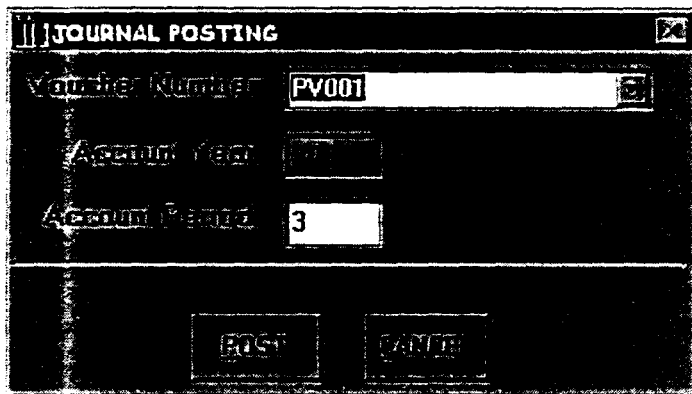
The Budget control management allows for the analysis of account in the Vote book.

Actual		Budget	
Jan			
Feb			
Mar			
Apr			
May			
Jun			
Jul			
Aug			
Sep			
Oct			
Nov			
Dec			

**Form Name: FRMPOST.FRM**

This form allows a user to post a transaction. A Journal transaction is of no consequence if not ultimately posted – just like the manual processing. Posting of Journal transaction is recommended

to be performed by an accounting officer that did not enter such ledger. This is expected to provide a check against errors made maliciously or otherwise.



The screenshot shows a window titled "JOURNAL POSTING". Inside the window, there are three input fields. The first is labeled "Voucher Number" and contains the text "PV001". The second is labeled "Account Year" and contains the text "07". The third is labeled "Account Period" and contains the text "3". Below these fields, there are two buttons: "POST" on the left and "CANCEL" on the right.

For posting, it is required to supply the Voucher number and the Account Period, but the Account year is not optional, it has to be during the accounting year (system date year). Posting can be made into any account period within the accounting year.

Posting can only be made of current Journals. All current Journals are added to the list of voucher numbers in the combo box requesting the voucher number. Apart from selecting the voucher numbers one at a time, one can Post all pending current vouchers at a time by selecting "ALL" from the voucher number list.

If there are errors while posting, the Journal that causes the error is not posted.

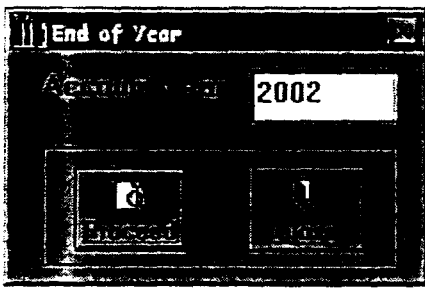
**WARNING** It should be noted that posting is not reversible.

**Form Name: FRMCONFIRM.FRM**

**Screen Purpose:** This form is used to initiate the end of year financial operation. This routine is supposed to be the last operation of a financial year. The routine automatically writes and updates records in readiness of the next accounting year.

The process involved here is the computation of Closing Balances for the current Accounting year, which is written onto the database as the Opening Balance for the next accounting year.

This process is also not reversible and must be carried out only when necessary.



### 3.6 OUTPUT DESIGN

Output Design (REPORTS as called in Visual Basic) is also important in any system development process. A good report is a basic instrument for management decision making, thus for a report to be alright it has to contain all bit of information required in it. The following output designs are used.



**Report Name: Budget.rpt**

**Report Purpose:** This report displays the Budgeted amount of accounts per annum. All accounts that is budgeted for are displayed in this report.

Account Code	Account Name	Budget (M)
101	CASH & BANK	256,321,322.00

**Report Name: Gledger.rpt**

**Report Purpose:** This report displays the ledger transaction activities at any point in time. The activities of the ledger can be displayed even after posting it (History), or if it is not yet posted (Current) or both (i.e. locates the journal if posted or not).

If display option is 'History' the voucher number combo only displays the voucher numbers of posted ledger. If display option is 'Current', the voucher number combo only displays the voucher numbers of ledgers not yet posted, but when the 'Both' option is chosen, all transacted vouchers are displayed.

**General Ledger Activity**

Voucher No: JV001

Journal Type: JV - Journal Voucher

Date FROM: 06-Mar-2001

Date TO: 06-Mar-2001

Asset  
 Liabilities  
 Both

Report Name: VoteBook.Rpt

*Report Purpose:* The Vote Book report analysis and displays the budget of an account with the current amount expended by the account.

**Vote Book (Summary)**

Account Year: 2001

Account type: 1 - Asset

The Vote book report requires the supply the Account year and Account type.

ALU-JIDZ LIMITED				
Print Date 07/Mar/2001		<u>Vote Book (Summary)</u>		
Account Type Asset				
Accounting Year 2001				
Account Code	Name	Amount (N)	Amount Balance (N)	Balance (N)
10201	Motor Vehicles	1,910,000.00		1,910,000.00
10202	Walkie Talkie Radio	362,616.00		362,616.00
10203	Furniture, Fittings + Equipments	349,958.00		349,958.00
10204	Furniture + Fittings (Household)	439,700.00		439,700.00

**Report Name:** *Audit.Rpt*

**Report Purpose:** This report displays a particular account transaction activity at any point in time. The account activities can be displayed even after posting it (History), or if it is not yet posted (Current) or both (i.e. locates the account if posted or not).

To display this report the Account Code number is necessary and a date range is needed to help the search

<b>Audit Trail</b>						
Account Code	10202 - Walkie Talkie Radio					
Date FROM	07-Mar-2001	<input checked="" type="radio"/> History				
Date TO	07-Mar-2001	<input type="radio"/> Current				
		<input type="radio"/> Both				
<table border="1"> <tr> <td> Preview</td> <td> Print</td> <td> Close</td> </tr> </table>				Preview	Print	Close
Preview	Print	Close				

ALU - LIMITED						
Print Date 07/Mar/2001		Audit Trail For Period Between 7/3/1942 and 7/3/2001				
(LEDGER)						
Account Code 10202						
Account Name Walkie Talkie Radio						
Date	Acct. Year	Period	LN	Description	DR	CR
28/Feb/2001	2001	2	FY002	Bought	382,616.00	
06/Jan/2001	2001	1	FY001	JY002 Chairman's Donation		15,000,000.00
					382,616.00	15,000,000.00
						14,637,384.00 Cr

**Report Name:** IncomeStat.rpt

**Report Purpose:** This report analysis income/revenue and expenditure using the budget index to determine the variance between the income/revenue, expenditure and the budget within a particular date range. To display this report the Account year, Account period and Account level required.

**Balances**

Account Year: 2001

Account Period: 3

2 - Details

Review Print Close

**ABUS DEBENTURES**

Print Date: 02/08/2001

**INCOME STATEMENT: BUDGET VARIANCE FOR PERIOD 3, 2001**

2 - Details

Account Code	Description	Actual	Budget	Variance	Actual	Budget	Variance
<b>Income/Revenue</b>							
0001	Charitable Capital	0.00	0.00	0.00	0.00	0.00	0.00
0002	Share Holders Capital	0.00	0.00	0.00	0.00	0.00	0.00
0003	Donations	0.00	0.00	0.00	1,000,000.00	0.00	1,000,000.00
<b>Total: Income/Revenue</b>		<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>1,000,000.00</b>	<b>0.00</b>	<b>1,000,000.00</b>
<b>Expenditures</b>							
0101	Salaries & Wages	0.00	0.00	0.00	0.00	0.00	0.00
0102	Pharmacy	0.00	0.00	0.00	42,111,111.00	0.00	42,111,111.00
0103	Type setting	0.00	0.00	0.00	0.00	0.00	0.00
<b>Total: Expenditure</b>		<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>42,111,111.00</b>	<b>0.00</b>	<b>42,111,111.00</b>

Report Name: ProfitLoss.Rpt

This report generate the profit and lost account using income/revenue and expenditure accounts. To display this report the Account year, Account period and Account level required.

**Balances**

Account Year: 2001

Account Period: 3

2 - Details

Review Print Close

ALU REPORT NUMBER			
PROFIT & LOSS ACCOUNT AS AT 07-Mar-2001			
Print Date 07-03-01			
	(-N)	(-N)	(-N)
<b>INCOME/REVENUE</b>			
Chairman Capital		(2,000,000.00)	
Donations		(1,320,000.00)	
<b>Total INCOME/REVENUE</b>		<b>(3,320,000.00)</b>	
<b>EXPENDITURE</b>			
Photocopy	(42,111,111.00)		
<b>Total EXPENDITURE</b>	<b>(42,111,111.00)</b>		
		<b>Net Profit/Loss</b>	<b>38,291,111.00</b>

**Report Name:** TrialBal.Rpt

**Report Purpose:** This report generates the trial balance of transaction of the specified account year, account period and the account level. To display this report the Account year, Account period and Account level required.

**ABC - DIXIELIMITED**

Print Date 02/26/2001

**TRIAL BALANCE AS AT PERIOD 3, 2001**

Account Code	Description	Debit	Credit	Debit	Credit	Debit	Credit
1101	CASH	0.00	0.00	0.00	5,475,600.00	0.00	7,475,600.00
1101	Motor Vehicles	0.00	0.00	3,300,000.00	0.00	3,300,000.00	0.00
1102	Walter Latta Bldg	0.00	0.00	725,312.00	0.00	725,312.00	0.00
1103	Furniture, Fixtures & Equipment	0.00	0.00	2,549,875.00	0.00	2,549,875.00	0.00
1104	Furniture & Fixtures (Miscellaneous)	0.00	0.00	2,198,500.00	0.00	2,198,500.00	0.00
1104	Medical Payments	0.00	0.00	0.00	112,400,872.00	0.00	112,400,872.00
1102	Staff Accounts	0.00	0.00	0.00	0.00	0.00	0.00
1103	Prepayments-Rent	0.00	0.00	0.00	1,663,500.00	0.00	1,663,500.00
1104	Edison Bldg-PLC	0.00	0.00	0.00	0.00	0.00	0.00
1104	Julius Bremer Remittance	0.00	0.00	0.00	0.00	0.00	0.00
1106	VAT Input A/C	0.00	0.00	0.00	0.00	0.00	0.00
1107	Bank-Cash	0.00	0.00	0.00	0.00	0.00	0.00
1108	IBB	0.00	0.00	90,378,602.00	0.00	90,378,602.00	0.00
1108	Staff Credits	0.00	0.00	12,518,110.00	0.00	12,518,110.00	0.00
1109	Credit for Share Capital	0.00	0.00	0.00	1,833,781.00	0.00	1,833,781.00
1102	Profit & Loss Account (Default)	0.00	0.00	56,944,644.00	0.00	56,944,644.00	0.00
1101	Share Capital & Premium	0.00	0.00	0.00	0.00	0.00	0.00
1102	Provisionary	0.00	0.00	0.00	42,111,111.00	0.00	42,111,111.00
1103	Type setting	0.00	0.00	0.00	0.00	0.00	0.00
1101	Common Credit	0.00	0.00	0.00	2,000,000.00	0.00	2,000,000.00
1102	Shareholder Credit	0.00	0.00	0.00	0.00	0.00	0.00
1103	Dividends	0.00	0.00	1,000,000.00	0.00	1,000,000.00	0.00
1110	Liability A/C	0.00	0.00	10,668,110.00	0.00	10,668,110.00	0.00
		0.00	0.00	179,348,872.00	179,348,872.00	179,348,872.00	179,348,872.00

Report Name: BalanceSheet.Rpt

Report Purpose: This report generates the trial balance of transaction of the specified account year, account period and the account level. To display this report the Account year, Account period and Account level required.

**Balance** [X]

Account Year:  Account Level:

Account Period:

**ALU IDIZ LIMITED**

**BALANCE SHEET AS AT 07-Mar-2001**

Print Date 07-03-01

	(=N=)	(=N=)	(=N=)
<b><u>FIXED ASSETS</u></b>			
Motor Vehicles		3,820,000.00	
Walkie Talkie Radio		725,232.00	
Furniture, Fittings + Equipments		2,549,874.00	
Furniture + Fittings (Household)		2,198,500.00	
<b>Total FIXED ASSETS</b>			9,293,606.00
<b><u>CURRENT ASSETS</u></b>			
NHF		90,878,602.60	
Staff Creditors		12,518,110.00	
Contact Advance		10,668,110.00	
<b>Total CURRENT ASSETS</b>			114,064,822.60
<b><u>CURRENT LIABILITIES</u></b>			
CASH	(5,473,606.00)		
Work-In-Progress	(112,400,872.60)		
Prepayment-Rent	(1,663,500.00)		
<b>Total CURRENT LIABILITIES</b>			(119,537,978.60)
<b><u>CAPITAL AND RESERVE</u></b>			
Called up Share Capital		(4,833,783.00)	
Profit & Loss Account (Deficit)		46,944,444.00	
<b>Total CAPITAL AND RESERVE</b>			42,110,661.00



## CHAPTER FOUR

### SYSTEM IMPLEMENTATION

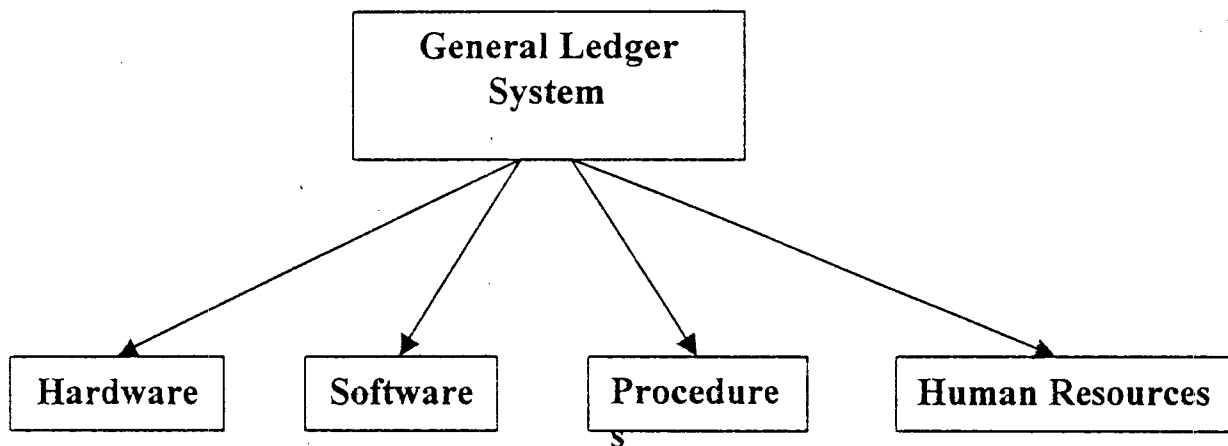
#### 4.1 INTRODUCTION

Implementation is the process of applying the developed system for the purpose it is meant for. System implementation involves the development of quality assurance procedures, including data security, back-up, recovery and system control system implementation objective is to complete the orderly and unobtrusive installation of the new system. During the system implementation, the new system is installed and users have the opportunity to operate the new system in “parallel” with the existing system.

The system implementation comprises the following task:

- Application system installation
- Documentation to provide user manuals
- Users Training on the new system
- Parallel system testing
- Data conversion/migration
- Acceptance of Testing
- System setup

#### 4.2 REQUIREMENTS FOR IMPLEMENTING A COMPUTERIZED LEDGER SYSTEMS



##### 4.2.1 HARDWARE REQUIREMENT

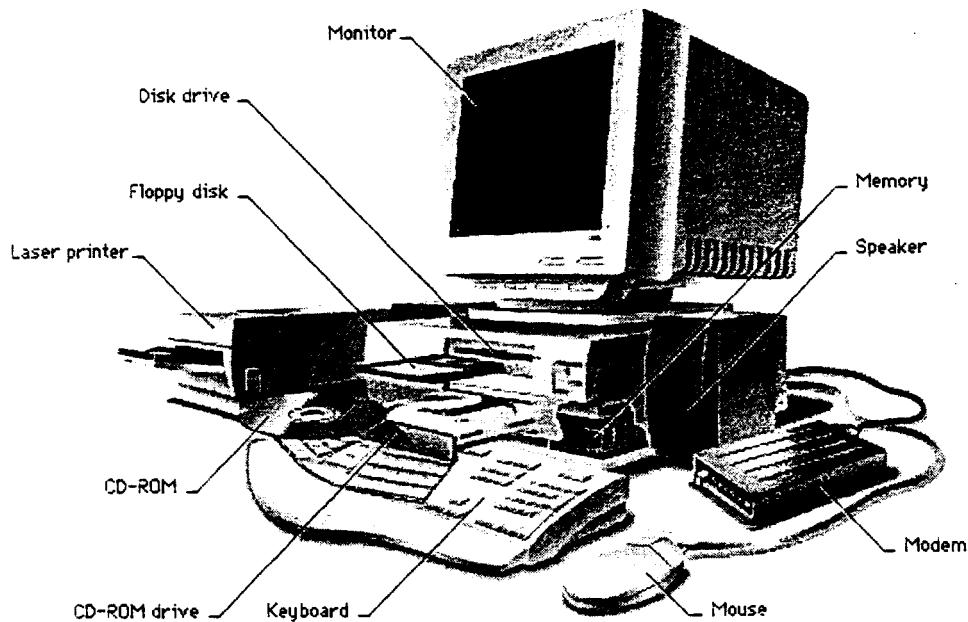
Computer hardware consists of the components that can be physically handled. The function of these components is typically divided into three main categories: input, output and storage. Components in

these categories connect to microprocessors, specifically, the computer's Central Processing Unit (CPU), which is the electronic circuitry, that provides the computerized ability and control of the computer, via wires or circuitries.

In recent years microcomputers have been designed to pack within them more computing powers, in years past this was the exclusive preserve of minis and mainframes, but it is now possible to use microcomputers (as a Stand-alone or in a Network) for more demanding task.

For Ledger Control System, a complete Microcomputer System would be ideal. Such a system should also have along, a Printer for the production of hardcopy.

## **Computer Hardware Components**



© Microsoft Corporation. All Rights Reserved.

### **Hardware Configuration**

#### **Full System**

- Pentium PIII 550 Processor
- 6.4 GB Hard Disk
- 15" Digital Color Monitor
- ATX Mini Tower Casing
- 120-Watts Speakers (Optional)
- 54x CD-ROM (Optional)
- 64 MB RAM

#### **Printer**

DeskJet or Epson Line Printer

#### **UPS**

APC or Sun Pac UPS (640KV)

### **4.2.2 SOFTWARE REQUIREMENT**

Software are computer programs – instructions that cause the hardware – the machine – to do work. Software as a whole can be divided into a number of categories based on the types of work done by

the programs. Two primary software categories are Operating System (System Software), which control the workings of the computer and Application Software which addresses the multitude of tasks for which people use computers. System software thus handles such essential, but often invisible chores as maintaining disk files and managing the screen (display), whereas Application software perform word processing, database management, and the like. Two additional categories that are neither System nor Application software, although they contain elements of both, are network software, which enables groups of computers to communicate, and Language software, which provide programmers with the tools they need to write programs.

For a Ledger system, the System software (Operating System) required is the Windows Operating System, while the Application software can either be:

#### *Customized Ledger Automation Software*

This is a software written with the sole aim of managing a ledger system. It is referred to as customized because it is particularly tailored to meet the specific need or requirement of the organization, ministry, or state that requires it. It can be gotten by commissioning a team of programmers (consultants) or qualified computer analyst to understudy the manual system and design computer application software that would simulate the manual procedures.

In doing this cognizance is always taken of the:

- Need for the software to reflect effectively, the operations of the manual procedures.
- User friendliness - such software should be easy for the user to use
- Eliminate the shortcoming of the manual procedures
- Reliability – meet the design objective

The objective of this project is to develop this type of Application software to replace the manual mode of general ledger management

### **4.2.3 PROCEDURES**

Procedures are step-by-step method(s) of using a system to be able to achieve result. A procedure in the context of this project is a physical component because they are provided in a physical form such as manual and instruction booklets. The major types of procedures that are required are:

- User instructions
- Instruction for preparation of input
- Operating instructions for the computer center personnel.

#### **4.2.4 PEOPLE (HUMAN RESOURCES)**

Human resources are the most important of any data processing system, because it is its responsibility to control and coordinate all other component that makes up the entire system.

It is necessary to have qualified personnel whose responsibility it will be to manage the system. Such personnel needs to be trained effectively on:

- The use of the computer.
- Manipulation of office systems.
- Use and operation of database software to be used

#### **4.3 SYSTEM TESTING**

After the installation of the new system, the system must undergo a test, once all the programs have been written and the training of the personnel to use the system is completed. The system testing is to ensure that all the sub-programs have been efficiently and correctly written. The system testing entails the execution of the program with test data so as to enable the system developer and the management to know the operational efficiency of the system.

The system testing will also enable the designer to correct errors and delete programs or modules that are not efficient or relevant by a process called debugging, using test data input into the programs so as to produce the desired output reports. Test data of all possible type/kind are used in other that all likely behaviors of the system to the input is ascertained before actual system implementation.

During this task, the Programmers or the System designer(s) assists the project staff in conducting the testing of the developed system so as to ensure that the system meets all the users needs and requirements. System testing entails the testing and certification of the system developed. This phase ensures that all required features, functions and capabilities are present in the system developed, and that all other requirements are met. Any necessary revisions are made during the system testing.

It is note-worthy that test data should be of 'real-live' nature.

#### **4.4 SYSTEM SET-UP**

After the successful System Testing, and the system output or requirements are mutually accepted by the Users of the system, the System Analyst and System Programmer(s), the next thing in the system

implementation line is the setting up of the installed system, that is, putting in place or entering the basic information necessary for the system smooth take off.

Setup information in this new system include:

- **Authorization Setup:** Here it is required to identify the users of the new system and Password assigned to them. It is also note-worthy that not every user can have unlimited access to the entire system, thus, access levels would also be determined for each user of the system. Authorization need not be done every time the system is ran, but only when new users are to be given access to the system or if modification is necessary – changing password or access level. The assignment of access to users can only be done be the System Administrator.
- **Table Maintenance:** Frequently used information are kept in a database table so that they are not always supplied by the users, example is when entering information of personnel, such information like the State of Origin, Designation, Grade Levels are usually within a known range, therefore they are pre-entered into a table so that users entering such data would only need to pick/select from the list.
- **User Information:** The new system is an ‘open-system’, and can be used by any other organization or company that uses or want to use the same method in this system for its general ledger management. Thus it is pertinent that the user information is supplied once, upon the first running of the system.

#### 4.5 CHANGE-OVER & DATA MIGRATION

The change over from old to new system may take place when the system has been proved to the satisfaction of the new System Analyst and the other implementation activities have been completed. The method and approach used for the change over is the parallel running system. The parallel system testing means processing current data by both the old and new system concurrently, to cross check the result and compares them. The main advantage is that the old system is kept alive and operational until the new system has been proved for at least one system circle. Using ‘real-live’ data in the real operational environment of the equipments, people and data, the results of the new system will be compared with old system to ensure the efficiency, capability and durability before acceptance by the user.

The change over task is designed to ensure that the software developed replicate the functionality of the system to be replaced.

Once the change over ends, the user staff complete their training and the parallel system testing are successful, the conversion of records of the old system to the file format of the new system, which involve data entry/capture of several forms of data using the software (*Data Migration*), is necessary.

#### 4.6 SYSTEM INSTALLATION

Ledger<sup>+</sup> comes either on Diskettes or CD-ROM, the installation of Ledger<sup>+</sup> is similar to any major installation of applications developed for Windows 9x Operating System e.g. Microsoft Office. A wizard takes you through the installation process, on strict adherence to the instructions or questions needed to complete the installation process, then the installation would go without a hitch.

**NOTE** it is required at a point, when asked, to change the Directory which Ledger<sup>+</sup> would be installed to C:\MHLedger, because by default the wizard would want to install it in the Programme Folder.

Ledger<sup>+</sup> Installation provides for re-installation, but to keep the old database for use by the program it is required to back it up and restore it into the C:\MHLedger Folder after the re-installation, otherwise the Old database would be over written during the re-installation. Just like any other windows based software Ledger<sup>+</sup> can also be un-installed using the appropriate Uninstall facility in the Operating System.

#### SYSTEM SETUP AND CONFIGURATION

This section explains how to install and setup the General Ledger Accounting Management Software for your particular use. There are a few things that must be configured for your use in other to help you correctly enter data into the databases.

**WARNING:** If you make changes to any database used in a report after previous reports have been ran. The reports will no longer match.

The following items need to be setup before entering student information into the database:

#### 4.7 ACCOUNT TYPE DEFINITION

The Account Type setup is a requirement for the running of the program. The software requires to know the various types of accounts to be transacted, hence it determines the various transact-able accounts positions in some reports like Balance Sheet.

Account setup must be defined as types:

	<i>Code</i>		<i>Description</i>
(i)	1	-	Fixed Asset
(ii)	2	-	Current Assets or Current Liability
(iii)	3	-	Income/Revenue
(iv)	4	-	Expenditure
(v)	5	-	Capital & Reserves

It should be noted that a definition order than this might cause abnormal behavior of the entire system. Having this in mind, it is thus pertinent that this setup is shipped with the package, and the option of changing the Account Type discouraged by disabling this option in the main program.

#### 4.8 ACCOUNT LEVEL DEFINITION

The Account Level configuration is another important procedure that must be undertaken if the software is to function well.

The Account Level determines the category of transact-able item/accounts in the General Ledger. The Software defaults any account into either of two levels viz:

	<i>Code</i>		<i>Description</i>
(i)	1	-	General
(ii)	2	-	Detail

**General Account** is the Control Account in the Chart of Account, the accounts with a General (1) Level is a Heading to other Detail accounts; thus, a General account cannot be transacted directly. The General account accumulates the various transactions of all Detail accounts it controls.

**Detail Account** is an account that can be transacted directly, all detail accounts **MUST** have a Control account which is a General account. The Detail accounts are expected to be more than the General accounts.

Like the Account Type setup, the Account Level is also restricted to this two levels and thus changes is impossible directly except by with the programmer's involvement.



## **CHAPTER FIVE: CONCLUSION AND RECOMMENDATION**

### **5.1 CONCLUSION**

The automated general ledger system is an indispensable tool for a complete accounting system management especially in Nigeria. With the crave for technological advancement in the entire world, the appropriate technological tool such as this Ledger Management system should be highly embraces and applied.

The success of any organisation is dependent on how well such organisation is managed. Management in turn relies on the information supplied to it to make decisions. Thus the use of an effective General Ledger System that shows the proper accounting status of organisation finance would be very much desirable. This because it would help in no small way in enhance the organisation efficiency and its total well-being.

### **5.1 RECOMMENDATION**

The automated general ledger system is highly recommended for use by government agency, corporations and organization and small and medium scale businesses.

In today's business there is a gradual transition from manual to electronic information management. This trend has also affected the accounting system. Most manual accounting procedure are how being handled use computer and other automated accounting devices. Therefore the automated general ledger system would be most useful for most business.

For further development the automated ledger system can be redesigned to incorporate such modules as;

- Cash Book Analysis.
- Account receivable and Payable.
- Investments.
- Asset Classification and Management.

The inclusion of these would make it a total and complete accounting package.

## References

---

- Buckley, John W. and Buckley, Marlene H. *The Accounting Profession*. Wiley, 1974. Books on Demand UMI., History, associations, education, and changes.
- Carey, John L. and Skousen, K. Fred. *Getting Acquainted with Accounting*. Houghton, 2d ed., 1977. Introduction to terms and tasks.
- Dyer, Mary Lee. *Practical Bookkeeping for the Small Business*. Contemporary, 1976. Simple book.
- Edwards, James D. and others. *How Accounting Works*. Dow Jones, 1983. "A guide for the perplexed" .
- Haller, Leon. *Making Sense of Accounting Information*. Van Nostrand, 1985. "A practical guide to understanding financial reports and their use"
- Handbook of Modern Accounting*. McGraw, 3d ed., 1983. Standard reference for the profession.
- Kohler's *Dictionary for Accountants*. Prentice, 6th ed., 1983. Handy for beginners and as a refresher; comprehensive.
- Meyer, Philip E., Author of *Applied Accounting Theory: A Financial Reporting Perspective*.
- Myer, John N. *Accounting for Non-Accountants*. Dutton, rev., 1987. For the businessperson with little accounting background; first published 1957.
- Sheff, Alexander L. *Bookkeeping Made Easy*. Harper, 1971. Introduction that includes business mathematics; first published 1938.

Stevens, Mark. *The Big Eight*. Macmillan, 1981. "Inside America's largest accounting firms"

Tracy, John A. *How to Read a Financial Report*. Wiley, 2d ed., 1983. "Wringing cash flow and other vital signs out of the numbers" .

"Accounting and Bookkeeping," *Microsoft® Encarta® Encyclopedia 99*. © 1993-1998  
Microsoft Corporation. All rights reserved.

**Public TheTitle As String**

**Private Sub Cmd1New\_Click()**

On Error GoTo handler  
Data1.Recordset.AddNew  
Text1.Enabled = True  
Text2.Enabled = True  
Cmd1Save.Enabled = True  
Cmd1Delete.Enabled = False  
Cmd1New.Enabled = False  
Text1.SetFocus  
Exit Sub  
handler:  
MsgBox Err.Description  
**End Sub**

**Private Sub Cmd1Save\_Click()**

On Error GoTo handler  
Data1.UpdateRecord  
Data1.Recordset.Bookmark =  
Data1.Recordset.LastModified  
Text1.Enabled = False  
Text2.Enabled = False  
Cmd1Save.Enabled = False  
Cmd1Delete.Enabled = True  
Cmd1New.Enabled = True  
Exit Sub  
handler:  
MsgBox Err.Description  
**End Sub**

**Private Sub Cmd1Refresh\_Click()**

On Error GoTo handler  
If Data1.Recordset.EditMode = 1 Then  
Data1.Recordset.Cancel  
End If  
Text1.Enabled = True  
Text2.Enabled = True  
Data1.Refresh  
Cmd1New.Enabled = True  
Cmd1Save.Enabled = False  
Exit Sub  
handler:  
MsgBox Err.Description  
**End Sub**

**Private Sub Cmd1Delete\_Click()**

On Error GoTo handler  
resp = MsgBox("Deletion of the current record" + Chr(13) +  
"Continue (y/n)", vbYesNo + vbCritical)  
If resp = vbYes Then  
Data1.Recordset.Delete  
Data1.Refresh  
End If  
Exit Sub  
handler:  
MsgBox Err.Description

**End Sub**

**Private Sub CmdClose\_Click()**

Unload Me  
**End Sub**

**Private Sub CmdFirst\_Click()**

On Error Resume Next  
Data1.Recordset.MoveFirst  
**End Sub**

**Private Sub CmdLast\_Click()**

On Error Resume Next  
Data1.Recordset.MoveLast  
**End Sub**

**Private Sub CmdNext\_Click()**

On Error Resume Next  
Data1.Recordset.MoveNext  
**End Sub**

**Private Sub CmdPrevious\_Click()**

On Error Resume Next  
Data1.Recordset.MovePrevious  
**End Sub**

**Private Sub Data1\_Reposition()**

Data1.Caption = Data1.Recordset.AbsolutePosition + 1  
**End Sub**

**Private Sub Form\_Load()**

Data1.DatabaseName = ThePath  
**End Sub**

**Private Sub mnuPreview\_Click()**

If UCase(Data1.RecordSource) = UCase("AcctLevel") Then  
CR1.ReportFileName = ThePath1 + "AcctLev.rpt"  
Else  
CR1.ReportFileName = ThePath1 + "AcctCode.rpt"  
End If  
CR1.Formulas(0) = "User=" & CoyName & ""  
CR1.Destination = crptToWindow  
CR1.WindowState = crptMaximized  
CR1.PrintReport  
**End Sub**

**Private Sub mnuPrint\_Click()**

If UCase(Data1.RecordSource) = UCase("AcctLevel") Then  
CR1.ReportFileName = ThePath1 + "AcctLev.rpt"  
Else  
CR1.ReportFileName = ThePath1 + "AcctCode.rpt"  
End If  
CR1.Formulas(0) = "User=" & CoyName & ""  
CR1.Destination = crptToPrinter  
CR1.PrintReport

End Sub

Private Sub Text1\_LostFocus()

If Not IsNumeric(Text1) Then

MsgBox "Invalid Entry a Number is required...",  
vbCritical

Text1 = ""

Text1.SetFocus

End If

End Sub

Private Sub CmdClose\_Click()

Unload Me

End Sub

Private Sub Combo1\_LostFocus()

If Combo1 <> "" Then

TheLen = InStr(Combo1, "-")

If TheLen <> 0 Then Combo1 = Trim(Mid\$(Combo1, 1,  
TheLen - 1))

End If

End Sub

Private Sub Command1\_Click()

On Error GoTo handler

TheStart = ""

TheEnd = ""

dbms.Execute ("DROP TABLE Trashbin")

If DTP1.Value <> "" Or DTP2.Value <> "" Then

If DTP1.Value = "" And DTP2.Value = "" Then

MsgBox "Incomplete Date entry...", vbInformation  
Exit Sub

Else

If DTP1.Value <> "" Then TheStart =  
Trim(Str(DTP1.Day)) + "/" + Trim(Str(DTP1.Month)) + "/" +  
Trim(Str(DTP1.Year))

If DTP2.Value <> "" Then TheEnd =  
Trim(Str(DTP2.Day)) + "/" + Trim(Str(DTP2.Month)) + "/" +  
Trim(Str(DTP2.Year))

End If

End If

SQL1 = ""

SQL2 = ""

SQL3 = ""

If Combo2 <> "" Then SQL1 = "JName = " &  
Trim(Combo2) & ""

If Combo1 <> "" Then SQL2 = "AcctJnlType = " &  
Trim(Combo1) & ""

If TheStart <> "" And TheEnd <> "" Then SQL3 = " Date  
>= #" & CDate(TheStart) & " #" & \_

" AND Date <= #" &

CDate(TheEnd) & " #"

MyQuery = ""

If SQL1 <> "" Then MyQuery = SQL1

If SQL2 <> "" Then

If MyQuery <> "" Then

MyQuery = MyQuery + " AND " + SQL2

Else

MyQuery = SQL2

End If

End If

If SQL3 <> "" Then

If MyQuery <> "" Then

MyQuery = MyQuery + " AND " + SQL3

Else

MyQuery = SQL3

End If

End If

If Option2.Value = True Then

dbms.Execute ("SELECT \* INTO TrashBin FROM JnlHist  
WHERE " + MyQuery)

Else

If Option1.Value = True Then

dbms.Execute ("SELECT \* INTO TrashBin FROM glJournal  
WHERE " + MyQuery)

Else

dbms.Execute ("SELECT \* INTO TrashBin FROM JnlHist  
WHERE " + MyQuery)

dbms.Execute ("INSERT INTO TrashBin SELECT \* FROM  
glJournal WHERE " + MyQuery)

End If

End If

Set Remi = dbms.OpenRecordset("SELECT \* FROM trashBin ")

If Remi.EOF = True Then

MsgBox "No Record of the given criteria", vbCritical

Exit Sub

End If

TheTitle = "Journal Activity For Period Between " + TheStart + "  
and " + TheEnd

TheTitle = "General Ledger Activity For Period Between " +  
TheStart + " and " + TheEnd

CR1.ReportFileName = ThePath1 + "JTrail.rpt"

CR1.Formulas(0) = "TheTitle=" & TheTitle & ""

CR1.Formulas(1) = "User=" & CoyName & ""

CR1.Destination = crptToWindow

If Mode = 2 Then CR1.Destination = crptToPrinter

CR1.WindowState = crptMaximized

CR1.PrintReport

Exit Sub

handler:

```

If Err.Number = 3376 Then
Resume Next
Err.Clear
Else
MsgBox Err.Description
End If
End Sub

```

```

Private Sub Command2_Click()
Mode = 2
Command1_Click
Mode = 1
End Sub

```

```

Private Sub Form_Load()
Data1.DatabaseName = ThePath
Option2_Click
Mode = 1
If Mode = 2 Then
Set rst1 = dbs.OpenRecordset("SELECT * FROM
JournalType")
Label4.Caption = "Journal Type"
While rst1.EOF = False
Combo1.AddItem rst1!Code + " - " + rst1!Desc
rst1.MoveNext
Wend

```

```

DTP1.Value = Now
DTP2.Value = Now
End Sub

```

```

Private Sub Option1_Click()
Combo2.Clear
Set rst1 = dbs.OpenRecordset("SELECT DISTINCT JName
FROM gjJournal")
While rst1.EOF = False
If Not IsNull(rst1!JName) Then Combo2.AddItem
rst1!JName
rst1.MoveNext
Wend
End Sub

```

```

Private Sub Option2_Click()
Combo2.Clear
Set rst1 = dbs.OpenRecordset("SELECT DISTINCT JName
FROM JnlHist")
While rst1.EOF = False
If Not IsNull(rst1!JName) Then Combo2.AddItem
rst1!JName
rst1.MoveNext
Wend
End Sub

```

```

Private Sub Option3_Click()
Combo2.Clear

```

```

Set rst1 = dbs.OpenRecordset("SELECT DISTINCT JName
FROM gjJournal ")
While rst1.EOF = False
Combo2.AddItem rst1!JName
rst1.MoveNext
Wend

```

```

Set rst1 = dbs.OpenRecordset("SELECT DISTINCT JName
FROM JnlHist")
While rst1.EOF = False
If Not IsNull(rst1!JName) Then Combo2.AddItem rst1!JName
rst1.MoveNext
Wend
End Sub

```

```

Dim TheType, Title As String
Private Sub CmdClose_Click()
Unload Me
End Sub

```

```

Private Sub Command1_Click()
On Error GoTo handler
Dim ram As Recordset
TheType = 0
If Combo1 <> "" Then
TheLen = InStr(Combo1, "-")
If TheLen <> 0 Then TheType = Trim(Mid$(Combo1, 1,
TheLen - 1))
End If

```

```

Select Case t
Case Is = 1
GetBalShe
Case Is = 2
GetlStat
Case Is = 3
GetTBal
Case Is = 4
GetIncome
End Select

```

```

CR1.Formulas(0) = "TheTitle=" & Title & ""
CR1.Formulas(1) = "Typ=" & Combo1 & ""
CR1.Formulas(2) = "User=" & CoyName & ""

```

```

CR1.Destination = crptToWindow
If Mode = 2 Then CR1.Destination = crptToPrinter
CR1.WindowState = crptMaximized
CR1.PrintReport

```

```

Exit Sub
handler:
MsgBox Err.Description, vbCritical

```

**End Sub**

**Private Sub Form\_Load()**

Text1 = Year(Now)

Text2 = Month(Now)

Mode = 1

Set rst1 = dbs.OpenRecordset("SELECT \* FROM  
AcctLevel")

While rst1.EOF = False

    Combo1.AddItem Str(rst1!Code) + " - " + rst1!Desc

    rst1.MoveNext

Wend

Combo1.ListIndex = (Combo1.ListCount) - 1

**End Sub**

**Public Sub GetTBal()**

On Error GoTo handler

dbs.Execute ("DROP TABLE Trash")

dbs.Execute ("DROP TABLE TrashBin")

Set OverAll = dbs.OpenRecordset("SELECT \* FROM  
glChart WHERE " & \_  
                                  " gllevel=" & TheType & """)

dbs.Execute ("SELECT \* INTO Trash FROM glLedger" &  
                                  " WHERE glYear=" & Text1)

dbs.Execute "CREATE TABLE TrashBin (CurBal  
CURRENCY," & \_  
                                  " PBal CURRENCY, glCode TEXT, glYear  
SINGLE,glName TEXT)"

Set Rat = dbs.OpenRecordset("SELECT \* FROM  
TrashBin")

Do While Not OverAll.EOF

Set ram = dbs.OpenRecordset("SELECT \* FROM glLedger  
WHERE glCode=" & OverAll.glCode & "" & \_  
                                  " AND glyear=" & Text1)

SQL1 = 0

If ram.EOF = False Then

For i = 1 To Text2 - 1

    If Not IsNull(ram.Fields(i).Value) Then SQL1 = SQL1

+ ram.Fields(i).Value

Next i

End If

Rat.AddNew

Rat!glCode = OverAll!glCode

Rat!glName = OverAll!glName

Rat!glYear = Text1

Rat!CurBal = SQL1

If ram.EOF = False Then

    Rat!PBal = ram.Fields(Val(Text2)).Value

Else

Rat!PBal = 0

End If

Rat.Update

OverAll.MoveNext

Loop

Title = "TRIAL BALANCE AS AT PERIOD " + Text2 + ", " +  
Text1

CR1.ReportFileName = ThePath1 + "tbal.rpt"

Exit Sub

handler:

If Err.Number = 3376 Then

    Resume Next

    Err.Clear

Else

    MsgBox Err.Description

End If

End Sub

**Public Sub GetIStat()**

On Error GoTo handler

dbs.Execute ("DROP TABLE Trash")

dbs.Execute ("DROP TABLE TrashBin")

Set OverAll = dbs.OpenRecordset("SELECT \* FROM glChart  
WHERE " & \_

                                  " gllevel=" & TheType & "" AND " & \_

                                  " (glType Like '3\*' OR glType Like '4\*')") '3

for income/revenue and 4 for Expenses

dbs.Execute ("SELECT \* INTO Trash FROM glLedger" & \_  
                                  " WHERE glYear=" & Text1)

dbs.Execute "CREATE TABLE TrashBin (CurBal CURRENCY," &  
                                  " \_  
                                  " YTODBudg CURRENCY,PBal CURRENCY,CBudg  
CURRENCY," & \_  
                                  " AcctType SINGLE, glCode TEXT, glYear  
SINGLE,glName TEXT)"

Set Rat = dbs.OpenRecordset("SELECT \* FROM TrashBin")

Do While Not OverAll.EOF

Set ram = dbs.OpenRecordset("SELECT \* FROM glLedger  
WHERE glCode=" & OverAll.glCode & "" & \_  
                                  " AND glyear=" & Text1)

SQL1 = 0

GetIt = 0

If ram.EOF = True Then GoTo ThisDay

For i = 1 To Text2 - 1

    If Not IsNull(ram.Fields(i).Value) Then SQL1 = SQL1 +  
ram.Fields(i).Value

```

Next i
GetIt = ram.Fields(Val(Text2)).Value
ThisDay:
Rat.AddNew
Rat!glCode = OverAll!glCode
Rat!glYear = Text1
Rat!CurBal = SQL1
Rat!PBal = GetIt
Rat!AcctType = OverAll!glType
Rat!glName = OverAll!glName
Set Dog = dbs.OpenRecordset("SELECT * FROM
glBudget WHERE glCode = " & OverAll!glCode & """)
SQL1 = 0
IDontKnowAgain = 0
Do While Not Dog.EOF
For i = 1 To Text2 - 1
If Not IsNull(Dog.Fields(i).Value) Then SQL1 =
SQL1 + Dog.Fields(i).Value
Next i
IDontKnowAgain =
IIf(IsNull(Dog.Fields(Val(Text2)).Value), 0,
Dog.Fields(Val(Text2)).Value)
Dog.MoveNext
Loop
Rat!YTODBudg = SQL1
Rat!Cbudg = IDontKnowAgain
Rat.Update
Another:
OverAll.MoveNext
Loop
Rat.Close

Title = "INCOME STATEMENT: BUDGET VARIANCE
FOR PERIOD " + Text2 + ", " + Text1

CR1.ReportFileName = ThePath1 + "IStat.rpt"

Exit Sub
handler:
If Err.Number = 3376 Then
Resume Next
Err.Clear
Else
MsgBox Err.Description
End If

End Sub

Public Sub GetBalShe1()
On Error GoTo handler

dbs.Execute ("DROP TABLE Trash")
dbs.Execute ("DROP TABLE TrashBin")

Set OverAll = dbs.OpenRecordset("SELECT * FROM
glChart WHERE " & _

```

```

" gllevel=" & TheType & " AND " & _
" (glType Like '1*' OR glType Like '2*' OR
glType Like '3*' OR glType Like '4*')") '1 for Asset, 2 for
Liability, 3 for Income/revenue and 4 for Expenses

dbs.Execute ("SELECT * INTO Trash FROM glLedger" & _
" WHERE glYear=" & Text1)

dbs.Execute "CREATE TABLE TrashBin (CurBal CURRENCY,"
& _
" AcctType SINGLE, glCode TEXT, glYear
SINGLE,glName TEXT)"

Set Rat = dbs.OpenRecordset("SELECT * FROM TrashBin")
Sum = 0

Do While Not OverAll.EOF
Set ram = dbs.OpenRecordset("SELECT * FROM glLedger
WHERE glCode=" & OverAll.glCode & "" & _
" AND glyear=" & Text1)

SQL1 = 0
GetIt = 0

If ram.EOF = True Then GoTo ThisDay
For i = 1 To Text2 - 1
If Not IsNull(ram.Fields(i).Value) Then SQL1 = SQL1 +
ram.Fields(i).Value
Next i
GetIt = ram.Fields(Val(Text2)).Value
SQL1 = SQL1 + GetIt + IIf(IsNull(ram.Fields(13).Value), 0,
IsNull(ram.Fields(13).Value))
ThisDay:
Rat.AddNew
Rat!glCode = OverAll!glCode
Rat!glYear = Text1
Rat!CurBal = SQL1
'Sum = Sum + IIf(overall!gltype Like "1*", SQL1,
IIf(overall!gltype Like "2*", -1 * SQL1, IIf(overall!gltype Like
"2*", -1 * SQL1, 0)))

'Rat!PBal = GetIt
Rat!AcctType = OverAll!glType
Rat!glName = OverAll!glName
Rat.Update

'Another:
OverAll.MoveNext
Loop
Rat.Close
Title = "BALANCE SHEET AS AT PERIOD " + Text2 + ", " +
Text1

CR1.ReportFileName = ThePath1 + "balShe.rpt"

Exit Sub

```



```

handler:
If Err.Number = 3376 Then
  Resume Next
  Err.Clear
Else
  MsgBox Err.Description
End If
End Sub

Public Sub GetBalShe()
On Error GoTo handler

```

'NOTE original Balance Sheet Computation is in procedure below i.e GetBalShe1

```

dbs.Execute ("DROP TABLE Trash")
dbs.Execute ("DROP TABLE TrashBin")

Set OverAll = dbs.OpenRecordset("SELECT * FROM
glChart WHERE " & _
  " gllevel=" & TheType & " AND " & _
  " (glType Like '1*' OR glType Like '2*'
OR glType Like '5*')") '1 for Asset, 2 for Liability, 5 for
Capital & Reserves

```

```

dbs.Execute ("SELECT * INTO Trash FROM glLedger" &
  " WHERE glYear=" & Text1)

```

```

dbs.Execute "CREATE TABLE TrashBin (CurBal
CURRENCY," & _
  "AcctType SINGLE, glCode TEXT, glYear
SINGLE,glName TEXT,Heading SINGLE)"

```

```

Set Rat = dbs.OpenRecordset("SELECT * FROM
TrashBin")
Sum = 0

```

```

Do While Not OverAll.EOF
Set ram = dbs.OpenRecordset("SELECT * FROM glLedger
WHERE glCode=" & OverAll.glCode & "" & _
  " AND glyear=" & Text1)

```

```

SQL1 = 0
GetIt = 0

```

```

If ram.EOF = True Then GoTo ThisDay
For i = 1 To Text2 - 1
  If Not IsNull(ram.Fields(i).Value) Then SQL1 = SQL1
+ ram.Fields(i).Value
Next i
GetIt = ram.Fields(Val(Text2)).Value

```

```

SQL1 = SQL1 + GetIt + IIf(IsNull(ram.Fields(13).Value), 0,
IsNull(ram.Fields(13).Value))
ThisDay:

```

```

If SQL1 = 0 Then GoTo Another 'don't put Zero entries

```

```

Rat.AddNew
Rat!glCode = OverAll!glCode
Rat!glYear = Text1
Rat!CurBal = SQL1
'Sum = Sum + IIf(overall!gltype Like "1*", SQL1,
IIf(overall!gltype Like "2*", -1 * SQL1, IIf(overall!gltype Like
"2*", -1 * SQL1, 0)))

```

```

'Rat!PBal = GetIt
Rat!AcctType = OverAll!glType
Rat!glName = OverAll!glName

```

```

'Defining the Arrangement
  If OverAll!glType = 1 Then Rat!Heading = 1 'For Fixed Asset
  If OverAll!glType = 2 And SQL1 >= 0 Then Rat!Heading = 2
'For Current Asset
  If OverAll!glType = 2 And SQL1 < 0 Then Rat!Heading = 3
'For Current Liabilities
  If OverAll!glType = 5 Then Rat!Heading = 4 'For Capital &
Reserves
  Rat.Update

```

```

Another:
OverAll.MoveNext
Loop

```

```

Rat.Close

```

```

Title = "BALANCE SHEET AS AT " & Format(Now, "dd-mmm-
yyyy")

```

```

CRI.ReportFileName = ThePath1 + "balShe1.rpt"

```

```

Exit Sub
handler:
If Err.Number = 3376 Then
  Resume Next
  Err.Clear
Else
  MsgBox Err.Description
End If

```

```

End Sub

```

```

Public Sub GetIncome()
On Error GoTo handler

```

```

dbs.Execute ("DROP TABLE Trash")
dbs.Execute ("DROP TABLE TrashBin")

Set OverAll = dbs.OpenRecordset("SELECT * FROM
glChart WHERE " & _
    " gllevel=" & TheType & " AND " & _
    " (glType Like '3*' OR glType Like
'4*')") '3 for Income/Revenue, 4 for Expenditure

dbs.Execute ("SELECT * INTO Trash FROM gLedger" &
    " WHERE glYear=" & Text1)

dbs.Execute "CREATE TABLE TrashBin (CurBal
CURRENCY," & _
    "AcctType SINGLE, glCode TEXT, glYear
SINGLE,glName TEXT,Heading SINGLE)"

Set Rat = dbs.OpenRecordset("SELECT * FROM
TrashBin")
Sum = 0

Do While Not OverAll.EOF
Set ram = dbs.OpenRecordset("SELECT * FROM gLedger
WHERE glCode=" & OverAll.glCode & "" & _
    " AND glyear=" & Text1)

SQL1 = 0
GetIt = 0

If ram.EOF = True Then GoTo ThisDay
For i = 1 To Text2 - 1
    If Not IsNull(ram.Fields(i).Value) Then SQL1 = SQL1
+ ram.Fields(i).Value
Next i
GetIt = ram.Fields(Val(Text2)).Value
SQL1 = SQL1 + GetIt +
IIf(IsNull(ram.Fields(13).Value), 0,
IsNull(ram.Fields(13).Value))
ThisDay:

If SQL1 = 0 Then GoTo Another 'don't put Zero entries

Rat.AddNew
Rat!glCode = OverAll!glCode
Rat!glYear = Text1
Rat!CurBal = SQL1
'Sum = Sum + IIf(overall!gltype Like "1*", SQL1,
IIf(overall!gltype Like "2*", -1 * SQL1, IIf(overall!gltype
Like "2*", -1 * SQL1, 0)))

Rat!PBal = GetIt
Rat!AcctType = OverAll!glType
Rat!glName = OverAll!glName

```

```

'Defining the Arrangement
    If OverAll!glType = 3 Then Rat!Heading = 1 'For Income
    If OverAll!glType = 4 Then Rat!Heading = 2 'For
Expenditure

Rat.Update

Another:
OverAll.MoveNext
Loop

Rat.Close

Title = "PROFIT & LOST ACCOUNT AS AT " & Format(Now,
"dd-mmm-yyyy")

CR1.ReportFileName = ThePath1 + "Inc.rpt"

Exit Sub
handler:
If Err.Number = 3376 Then
    Resume Next
    Err.Clear
Else
    MsgBox Err.Description
End If

End Sub

Dim rt1 As Recordset
Private Sub Combo1_LostFocus()
If Combo1 <> "" Then
    TheLen = InStr(Combo1, "-")
    If TheLen <> 0 Then Combo1 =
Trim(Mid$(Combo1, 1, TheLen - 1))
End If
End Sub

Private Sub Command1_Click()
On Error GoTo handler
Dim TheBal As Double
If UCase(Combo1) = "ALL" Then

Set rst = dbs.OpenRecordset("Select * FROM
GLJournal WHERE " & _
    " AcctYear=" & Text1 & " AND
Aperiod=" & Text2)
Else

```

```

Set rst = dbs.OpenRecordset("Select * FROM
GLJournal WHERE JName="" & Combo1 & ""
& _
        " AND AcctYear="" & Text1 & "
AND Aperiod="" & Text2)
End If
If rst.EOF = True Then
MsgBox "Records with The Journal Type not
found", vbCritical
Exit Sub
Else
resp = MsgBox("Continue Posting (y/n)?",
vbYesNo + vbInformation)
If resp = vbNo Then Exit Sub
End If
If rst.EOF = False Then rst.MoveLast
PB1.Min = 0
PB1.Max = rst.RecordCount
PB1.Visible = True
If rst.BOF = False Then rst.MoveFirst
Do While rst.EOF = False
PB1.Value = rst.AbsolutePosition
TheBal = 0
If IsNull(rst!Dr) Then
TheDR = 0
Else
TheDR = rst!Dr
End If

If IsNull(rst!Cr) Then
TheCR = 0
Else
TheCR = rst!Cr
End If
TheBal = TheDR - TheCR
Set rst2 = dbs.OpenRecordset("SELECT * FROM
gLedger WHERE glCode="" & rst!AcctCode &
"" & _
        " AND glYear="" &
rst!AcctYear)
If rst2.EOF = True Then
rst2.AddNew
rst2!glCode = rst!AcctCode
rst2!glYear = rst!AcctYear
Select Case rst!Aperiod
Case 1

```

```

rst2!glp01 = TheBal
Case 2
rst2!glp02 = TheBal
Case 3
rst2!glp03 = TheBal
Case 4
rst2!glp04 = TheBal
Case 5
rst2!glp05 = TheBal
Case 6
rst2!glp06 = TheBal
Case 7
rst2!glp07 = TheBal
Case 8
rst2!glp08 = TheBal
Case 9
rst2!glp09 = TheBal
Case 10
rst2!glp10 = TheBal
Case 11
rst2!glp11 = TheBal
Case 12
rst2!glp12 = TheBal
End Select
rst2.Update

Else
rst2.Edit
rst2!glCode = rst!AcctCode
rst2!glYear = rst!AcctYear
Select Case rst!Aperiod
Case 1
rst2!glp01 = rst2!glp01 + TheBal
Case 2
rst2!glp02 = rst2!glp02 + TheBal
Case 3
rst2!glp03 = rst2!glp03 + TheBal
Case 4
rst2!glp04 = rst2!glp04 + TheBal
Case 5
rst2!glp05 = rst2!glp05 + TheBal
Case 6
rst2!glp06 = rst2!glp06 + TheBal
Case 7
rst2!glp07 = rst2!glp07 + TheBal
Case 8

```

```

rst2!glp08 = rst2!glp08 + TheBal
Case 9
rst2!glp09 = rst2!glp09 + TheBal
Case 10
rst2!glp10 = rst2!glp10 + TheBal
Case 11
rst2!glp11 = rst2!glp11 + TheBal
Case 12
rst2!glp12 = rst2!glp12 + TheBal
End Select
rst2.Update
End If
rst.Edit
rst.Posted = True
rst.Update

Call CtrlAcct(rst!AcctCode, Text2, Text1,
TheBal)

rst.MoveNext
Loop
PB1.Visible = False

If UCase(Combo1) = "ALL" Then

TheQuery = "FROM GLJournal WHERE " & _
           " AcctYear=" & Text1 & " AND
Aperiod=" & Text2
Else
TheQuery = "FROM GLJournal WHERE
JName=" & Combo1 & "" & _
           " AND AcctYear=" & Text1 & "
AND Aperiod=" & Text2
End If

dbs.Execute ("INSERT INTO JnlHist SELECT *
" & TheQuery)
dbs.Execute ("DELETE * " & TheQuery)

MsgBox "Posting Completed...", vbInformation

```

```

Combo1 = ""
Combo1.Clear
Call Form_Load
Exit Sub
handler:
MsgBox Err.Description
End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
Set rt1 = dbs.OpenRecordset("SELECT DISTINCT
JName FROM GLJournal")
Combo1.AddItem "ALL"
While rt1.EOF = False
Combo1.AddItem rt1!JName
rt1.MoveNext
Wend
Text1 = Year(Now)
Text2 = Month(Now)
Combo1.ListIndex = 0
End Sub

Public Sub CtrlAcct(AcctNum, Period, ActYear,
Amt)
Set TestForCtrlAcct = dbs.OpenRecordset("SELECT
glControl FROM glChart WHERE glCode=" &
AcctNum & """)
'Dim CtrlAcct As Text

If TestForCtrlAcct.EOF = False Then
CtrlAcct1 = TestForCtrlAcct!glControl
Else
MsgBox "Control Account is not found for Account
" + AcctNum, vbCritical
Exit Sub
End If

If IsNull(TestForCtrlAcct!glControl) = True Then
MsgBox "Control Account is not found for Account
" + AcctNum, vbCritical
Exit Sub
End If

```

```

If CtrlAcct1 = AcctNum Then Exit Sub

Set rst2 = dbs.OpenRecordset("SELECT * FROM
gLedger WHERE glCode="" & CtrlAcct1 & "" &
-
"AND glYear=" & Text1)
If rst2.EOF = True Then

rst2.AddNew
rst2!glCode = CtrlAcct1
rst2!glYear = ActYear
Select Case Period
Case 1
rst2!glp01 = Amt
Case 2
rst2!glp02 = Amt
Case 3
rst2!glp03 = Amt
Case 4
rst2!glp04 = Amt
Case 5
rst2!glp05 = Amt
Case 6
rst2!glp06 = Amt
Case 7
rst2!glp07 = Amt
Case 8
rst2!glp08 = Amt
Case 9
rst2!glp09 = Amt
Case 10
rst2!glp10 = Amt
Case 11
rst2!glp11 = Amt
Case 12
rst2!glp12 = Amt
End Select
rst2.Update

Else
rst2.Edit
'rst2!GlCode = rst!AcctCode
'rst2!glYear = rst!AcctYear
Select Case Period
Case 1
rst2!glp01 = rst2!glp01 + Amt

```

```

Case 2
rst2!glp02 = rst2!glp02 + Amt
Case 3
rst2!glp03 = rst2!glp03 + Amt
Case 4
rst2!glp04 = rst2!glp04 + Amt
Case 5
rst2!glp05 = rst2!glp05 + Amt
Case 6
rst2!glp06 = rst2!glp06 + Amt
Case 7
rst2!glp07 = rst2!glp07 + Amt
Case 8
rst2!glp08 = rst2!glp08 + Amt
Case 9
rst2!glp09 = rst2!glp09 + Amt
Case 10
rst2!glp10 = rst2!glp10 + Amt
Case 11
rst2!glp11 = rst2!glp11 + Amt
Case 12
rst2!glp12 = rst2!glp12 + Amt
End Select

rst2.Update

End If

End Sub

Dim TheMode As Single
Dim InitialIndex As Long

Private Sub Combo1_Change()
If Combo1 = "" Then
DBGrid1.Enabled = False
Else
DBGrid1.Enabled = True
End If
End Sub

Private Sub Combo1_Click()
Combo1_Change
End Sub

Private Sub Combo1_LostFocus()

```

```

If Combo1 = "" Then
MsgBox "The Journal Type must be entered",
vbInformation
Combo1.SetFocus
Exit Sub
End If

If Combo1 <> "" Then
TheLen = InStr(Combo1, "-")
If TheLen <> 0 Then Combo1 =
Trim(Mid$(Combo1, 1, TheLen - 1))
End If

TheTab = "glJournal"
TheQuery = "Select max(acctSerial) AS SN
FROM "
TheQuery2 = " WHERE " & _
"AcctYear=" & Text1 & " AND
Aperiod=" & Text2 & _
" AND AcctJnlType=" &
Trim(Combo1) & """"
Set rst = dbs.OpenRecordset(TheQuery + TheTab
+ TheQuery2)

If Not IsNull(rst!SN) Then
t1 = rst!SN
Else
t1 = 0
End If
TheTab = "JnlHist"
Set rst = dbs.OpenRecordset(TheQuery + TheTab
+ TheQuery2)
If Not IsNull(rst!SN) Then
t2 = rst!SN
Else
t2 = 0
End If

If t1 >= t2 Then
TheIndex = t1 + 1
Else
TheIndex = t2 + 1
End If

```

```

InitialIndex = TheIndex
End Sub

```

```

Private Sub Combo2_LostFocus()
If Combo2 <> "" Then
TheLen = InStr(Combo2, "-")
If TheLen <> 0 Then Combo2 =
Trim(Mid$(Combo2, 1, TheLen - 1))
End If
Set cunle = dbs.OpenRecordset("SELECT glCode
FROM glchart WHERE " & _
" glcode=" & Combo2 & """" & _
" AND gilevel=" & ALevel & """")

```

```

If cunle.EOF = True Then
MsgBox "Account Code is Either General or Non-
existent", vbCritical
Combo2 = ""

```

```

End If
DBGrid1.Columns(3) = Trim(Combo2)
Combo2.Visible = False
End Sub

```

```

Private Sub Command1_Click()
dbs.Execute ("DELETE * FROM GJJournal WHERE
JName = " & Trim(Text3) & """" & _
" AND AcctYear=" & Trim(Text1) & "
AND Aperiod=" & Trim(Text2))
dbs.Execute ("DELETE * FROM JnlTrash")
DBGrid1.ReBind
End Sub
Private Sub Command2_Click()
Unload Me
End Sub

```

```

Private Sub Command3_Click()
If Command3.Caption = "&New" Then
Mode = 1
Temp = ""
Temp = InputBox("Specify Voucher Number",
"Voucher Number")

```

```

If Temp = "" Then Exit Sub

```

```

If Ck4Existence(Trim(Temp)) = True Then Exit Sub

```



```
Command3.Enabled = False
Else
Command3.Enabled = True
End If
End Sub
```

```
Public Function Ck4Existence(Vnum As
String) As Boolean
Ck4Existence = False
Set rst = dbs.OpenRecordset("SELECT JName
FROM Gjournal WHERE " &
"JName = " & Vnum & """)
If rst.EOF = False Then
MsgBox "Voucher Number already exist",
vbCritical
Ck4Existence = True
Exit Function
End If
```

```
Set rst = dbs.OpenRecordset("SELECT JName
FROM JnlHist WHERE " &
"JName = " & Vnum & """)
If rst.EOF = False Then
MsgBox "Voucher Number already exist and
Posted ", vbCritical
Ck4Existence = True
Exit Function
End If
```

```
End Function
```



```

Command3.Enabled = False
Command1.Enabled = True
DTP1.Value = rst!Date

dbs.Execute ("DELETE * FROM JnlTrash")
dbs.Execute ("INSERT INTO JnlTrash SELECT
* FROM Gjournal WHERE " & _
           "JName = " & Temp & """)
DBGrid1.Enabled = True
Data1.Refresh
DBGrid1.ReBind
DBGrid1.Refresh
DBGrid1_AfterUpdate
Else
MsgBox "Record not found...", vbCritical
End If

End Sub

Private Sub DBGrid1_AfterColEdit(ByVal
ColIndex As Integer)
If ColIndex = 3 Then
Set cunle = dbs.OpenRecordset("SELECT glCode
FROM glchart WHERE " & _
           " glcode=" & DBGrid1.Text &
           "" & _
           " AND gllevel=" & ALevel &
           """)
If cunle.EOF = True Then
MsgBox "Account Code is Either General or
Non-existent", vbCritical
DBGrid1.Text = ""
End If
End If
DBGrid1.Col = 0
DBGrid1.Text = TheIndex
End Sub

Private Sub DBGrid1_AfterUpdate()
dbs.Execute ("UPDATE JnlTrash SET
CR=NULL WHERE CR=0")
dbs.Execute ("UPDATE JnlTrash SET
DR=NULL WHERE DR=0")
Set rst1 = dbs.OpenRecordset("SELECT
SUM(Dr) AS TheDR, SUM(CR) AS TheCR
FROM JnlTrash")

```

```

If IsNull(rst1!TheCR) = True Then
TheCR = 0
Else
TheCR = rst1!TheCR
End If

If IsNull(rst1!TheDR) = True Then
TheDR = 0
Else
TheDR = rst1!TheDR
End If
Text4 = TheDR - TheCR
TheIndex = TheIndex + 1
Text5 = TheIndex
End Sub

Private Sub DBGrid1_BeforeColUpdate(ByVal
ColIndex As Integer, OldValue As Variant, Cancel As
Integer)
If ColIndex = 6 And DBGrid1.Text = "" Then
DBGrid1.Text = 0
End Sub

Private Sub DBGrid1_BeforeUpdate(Cancel As
Integer)
On Error GoTo handler
If Text1 <> "" And Text2 <> "" And Combo1 <> ""
Then
Data1.Recordset!AcctYear = Text1
Data1.Recordset!APeriod = Text2
Data1.Recordset!User = UserName
Data1.Recordset!JName = Text3
Data1.Recordset!AcctJnlType = Combo1
Data1.Recordset!Date = Format(DTP1.Value, "dd-mm-
yy")
TheIndex = TheIndex + 1
Else
MsgBox "Incomplete Data ...", vbInformation
Data1.Recordset.Cancel
Data1.Refresh
End If
Exit Sub
handler:
MsgBox Err.Description, vbCritical
End Sub

```

```

Private Sub DBGrid1_ButtonClick(ByVal
ColIndex As Integer)
Dim nRow As Byte
If ColIndex = 0 Then
nRow = DBGrid1.Row
If DBGrid1.Col = 0 Then
If ColIndex = 3 Then
'Combo2.Width = DBGrid1.Columns(3).Width
'Combo1.Height = DB1.RowHeight
Combo2.Left = DBGrid1.Columns(3).Left +
DBGrid1.Left
Combo2.Top = DBGrid1.RowTop(nRow) +
DBGrid1.Top
Combo2.Visible = True
Combo2.SetFocus
SendKeys "{F4}"
End If
DBGrid1.Col = 0
DBGrid1.Text = TheIndex
End Sub

```

```

Private Sub DBGrid1_Change()
DBGrid1_AfterUpdate
End Sub

```

```

Private Sub DBGrid1_Error(ByVal DataError
As Integer, Response As Integer)
If Err.Number = 16389 Then
Resume Next
Err.Clear
Else
MsgBox Err.Description
End If
End Sub

```

```

Private Sub DBGrid1_GotFocus()
If Text1 = "" Or Text2 = "" Or Text3 = "" Or
Combo1 = "" Then
DBGrid1.Enabled = False
Else
DBGrid1.Enabled = True
End If
End Sub

```

```

Private Sub Form_Activate()

```

```

DTP1.Value = Now
End Sub

```

```

Private Sub Form_Load()
TheMode = 0
DTP1.Value = Now
Data1.DatabaseName = ThePath
dbs.Execute ("Delete * from JnlTrash")
Set rst1 = dbs.OpenRecordset("SELECT * FROM
glchart WHERE gllevel=" & ALevel & """)
While rst1.EOF = False
Combo2.AddItem rst1!glCode + " - " + rst1!glName
rst1.MoveNext
Wend
Set rst1 = dbs.OpenRecordset("SELECT * FROM
JournalType")
While rst1.EOF = False
Combo1.AddItem rst1!Code + " - " + rst1!Desc
rst1.MoveNext
Wend
Text1 = Year(Now)
Text2 = Month(Now)

```

```

End Sub

```

```

Private Sub Text1_LostFocus()
If Text1 = "" Then
MsgBox "The Account Year must be entered",
vbInformation
Text1.SetFocus
End If
End Sub

```

```

Private Sub Text2_LostFocus()
If Text2 = "" Then
MsgBox "The Account Period must be entered",
vbInformation
Text2.SetFocus
End If
End Sub

```

```

Private Sub Text4_Change()
If Text4 <> 0 Then
Command1.Enabled = False

```