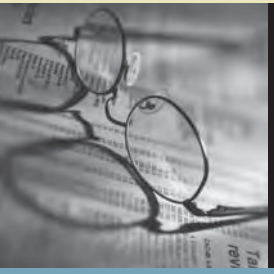


SUNGARD PUBLIC SECTOR

General Ledger

IFAS
Integrated Financial &
Administrative Solution

NOTICE

SUNGARD PUBLIC SECTOR BI-TECH LLC MAKES NO REPRESENTATIONS OR WARRANTIES, ORAL OR WRITTEN, EXPRESS OR IMPLIED, WITH RESPECT TO THE SYSTEM, SERVICES, SOFTWARE, DOCUMENTATION, OPERATING ENVIRONMENT, ANY OTHER SOFTWARE OR SERVICES PROVIDED HEREUNDER OR ANY OTHER MATTER ADDRESSED HEREUNDER, AND SUNGARD PUBLIC SECTOR BI-TECH LLC EXPLICITLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF TITLE, MERCHANTABILITY AND FITNESS FOR A SPECIFIC PURPOSE. SunGard Public Sector Bi-Tech LLC shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This documentation is proprietary and confidential information of SunGard Public Sector Bi-Tech LLC. Copying, reproduction or distribution is strictly prohibited. All rights reserved.

SunGard Public Sector Bi-Tech LLC

890 Fortress Street

Chico, CA 95973

Should you wish to make a comment, or if you find an error in the text, please contact us via email:

doc@bi-tech.com

Document Change Log

Version	Date	Change Description
7.9	April 2008	7.9 Version

Contents

1 System Overview

1.1 Introduction

The General Ledger is the focal point of all accounting functions. This ledger is used to keep track of and report all budget and actual information which relates to the organization. Most organizations are complex in terms of structure and management. Funding for activities comes from a wide range of sources. A number of people and entities have a need to know about the organization's accounting methods. This requires an organization to maintain financial information in different ways, depending upon the source and use of the funds. The reporting needs of an organization are also complex. There are several needs for commercial reporting for those aspects which are involved in revenue generation: for AICPA standard reports such as Statement of Net Assets, Statement of Net Activities, Income Statement, Balance Sheet, and Statement of Changes in Fund Balances; to report information to external agencies; to produce a variety of budget preparation reports; and, there is a never ending need to produce analytical information on demand for controlling bodies.

An organization accounts at three major levels: Cost Center, Fund, and Entity. The Cost Center is typically called a department, project, or responsibility area where we account for revenue and expenditures. The Fund level is where the assets and liabilities are managed for each of the given funds which make up the organization (e.g., General Fund, Capital Improvements Fund, etc.). The Entity level is where overall organization accounts are kept, such as cash and investment accounts. An organization needs to view budgeting and transactions from an operational perspective and perhaps from a Job Cost or Project Management perspective. On the operational side, the organization deals with line-item budgets for each department, project or cost center. Transactions are coded to these line-item budget accounts. On the Job Cost, or Project Management side, the organization needs to budget areas within a particular job or project. Transactions are then coded to these areas. The Job Cost or Project Management side can cross many of the operational accounts and vice verse. Due to the variety of sources and uses of funds, much of the organization's energy is involved in budget preparation and management.

The SunGard Public Sector IFAS General Ledger System is designed on the foundations set forth in AICPA Fund Accounting principles. It is intended for use by not-for-profit organizations. This system has been installed and implemented successfully in a wide range of not-for-profit settings: Colleges and universities, school districts, municipal governments, private and public foundations, auxiliary organizations, agencies, museums, educational TV stations, public works departments, blood banks, health related organizations, water districts, and many other organizations. There are two primary reasons one software package is able to meet the needs of such a disparate group of organizations. The organizations share a common need to account for unrestricted and restricted funding, and the software tailors itself to the specific characteristics of the client.

This user guide describes the General Ledger System. If you would like only to understand how SunGard Public Sector IFAS approaches the General Ledger, read only this overview section. To understand details required to use all aspects of the General Ledger, the entire user guide should be studied and used as a reference manual. Information regarding data entry, job dialogue, and reporting procedures is provided in *Getting Started User Guide*.

The remainder of the introductory section is intended to give the reader a "feel" for the organization of the General Ledger. First, the relationship of the General Ledger to other systems and the flow of data between them will be discussed. Second, the General Ledger database structure will be defined and a diagram will be presented which captures all theoretic aspects of the database structure. Third, the terminology used throughout our documentation is presented. Finally, a practical example is developed. The practical example is used throughout the remainder of this user guide to explain the use of the software. The reader must become familiar with the terminology and structure in order to fully understand how to take advantage of the many features included in this software package.

The General Ledger database is named "GLDB". It holds all budget and actual values which relate to the client's accounts. Each transaction held in the GLDB can reference two accounts: One is called the "GL Account" (this one is required on all transactions), and one is called the "JL Account" (this one is optional on a transaction). Thus, the GLDB is divided into two sides: The GL Side and the JL Side. Each transaction must reference the GL Side and may reference the JL Side.

The terms "GL Side" and "JL Side" are founded in the history of the development of the software package. A brief description of this history will provide you with a better understanding of what these two "sides" represent. The original version of the package, early 1970's, provided only one view of an account number, the "GL Side" and the interpretation of this account number was fixed by the AICPA format of "Fund, Function, Division, Cost Center, and Object Code" (i.e., these were the formal parts of a General Ledger account number). In the late 1970's, our clients began to ask for control over the definition of the formal parts of the General Ledger. The next version of the package permitted the client to define each of the desired formal parts and also permitted the client to use a shorter number which would stand for a particular combination of formal parts.

The next request from our clients came in the early 1980's. They wanted to use the package to form budgets and track activity which related to particular jobs or projects. They wanted all transactions to require a reference to the General Ledger structure and, when a transaction related to a particular job or project, they wanted the transaction to reference a given job or project account number. This additional account number was to be held in a transaction and became known as the Job Ledger account number. Our clients wanted the Job Ledger integrated with the General Ledger so that a transaction would only be entered into the system once and so that reconciliation between the two ledgers would be "foolproof".

As our client base grew and the reporting needs of our clients became more demanding, they began to request the ability to define different ledger formats within the "GL Side" and within the "JL Side". For example, the "GL Side" could be organized into one format for general operational activity and organized into another format for accounts relating to Agency activity. The "JL Side" could be organized into one format for funded grant activity and another format for Capital Improvement Projects.

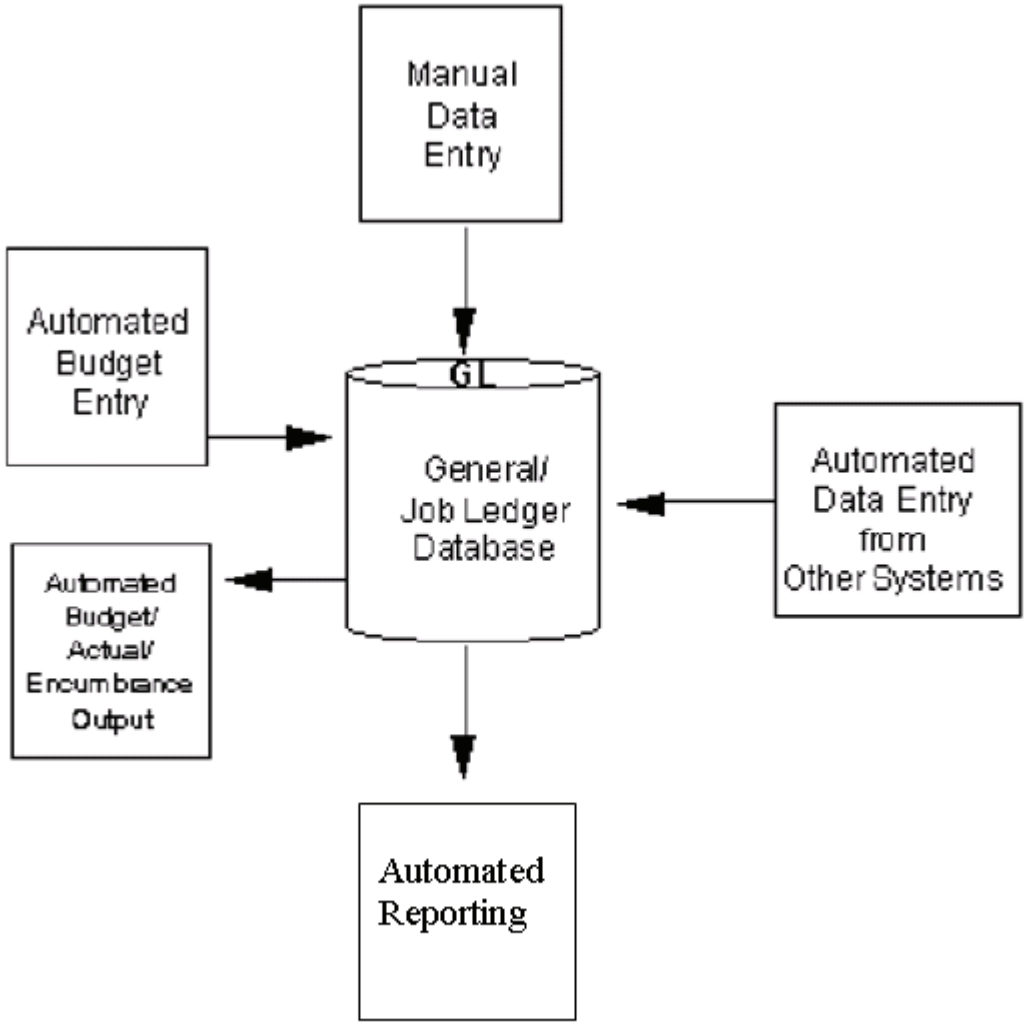
The history presented gives you a frame of reference for why we refer to a "GL Side" and a "JL Side". In a broad, theoretic sense, the naming of these sides is unimportant. The important concept is that each transaction can reference two account numbers, and that each is associated with a particular client defined ledger format. The diagram which follows will present this concept graphically.

1.2 System Flow Diagram

The purpose of this section is to provide you with information which describes the relationships among and within systems. The first diagram shows major functional modules (The Big Picture). In the second, specific points of integration between systems are discussed (The Integration Picture). Finally, subsequent diagrams show all system-to-system relationships with greater detail on functional modules (The Total Picture).

On the detail level diagrams, you will find *bold italic* characters. These characters relate to the codes which are entered in response to system dialogue and menu prompts (i.e., if the described characters are typed, the user will be taken to the diagrammed function).

1.2.1 The Big Picture



1.2.2 The Integration Picture

The following abbreviations are used in referencing the various systems within IFAS:

AP	Accounts Payable	AR	Accounts Receivable
BK	Bank Management	BQ	Bid Management
EN	Encumbrances	GL	General Ledger
FA	Fixed Assets	PC	Position Control
NU	Nucleus	PO	Purchasing
PE	Person/Entity	PY	Payroll
RX	Ad Hoc Report Writer	SY	System Level Software
BU	Budget	HR	Human Resources
SI	Stores Inventory	GM	Grant Management

GL - PO

- Interactive validation of GL/JL Account Numbers when a Purchase Request/Purchase Order is entered.
- Interactive budget checking when items are defined within Purchase Requests/Purchase Orders.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Update Encumbrance balances when Purchase Orders are printed.

GL - EN

- Interactive validation of account numbers from GL/JL when EN transactions are entered into a set.
- Interactive budget checking when EN transactions are entered into a set.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create new GL/JL Account (Org. Key/Object combination) if previously does not exist.
- Update Encumbrance balances when EN transactions are posted.
- GL On-line Inquiry and Standard Reports access EN transaction details and balances for reporting.
- EN transaction distribution may, at the client's discretion, cause transaction posting to the GL (this is rarely desired).

GL - AP

- Interactive validation of account numbers from GL/JL when AP transactions are entered into a set.
- Interactive budget checking when AP transactions are entered into a set.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create new GL/JL Account (Org. Key/Object combination) if previously does not exist.
- AP distribution posts GL/JL transactions and updates account balances with all automatic interfund and intrafund entries.
- Cash and accrual basis postings occur at client's discretion. Recurring Calculation process to create automatic AP transactions.

GL - CK

- Insert Check Number, Check Date, and IFAS Job Number in A/P expense transactions which are paid.
- Optional postings from Canceled Check Set to post from Check/Warrants Clearing to Cash.

GL - AR

- Interactive validation of account numbers from GL/JL when AR transactions are entered into a set.
- Optional, interactive budget checking against revenue accounts when AR transactions are entered into a set.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create new GL/JL Account (Org. Key/Object combination) if previously does not exist.
- Cash Receipts set supports both AR and non-AR related entries in one set.
- Cash and accrual basis postings occur at client's discretion.
- A/R and C/R distributions post GL/JL transactions and update account balances with automatic interfund and intrafund entries.
- Recurring Calculation process to create automatic AR transactions. Primarily used for reimbursement of grant activity.

GL - PE

- Extract name information to be reported with detail GL transactions.

GL - FA

- Depreciation and Write-off postings update GL transactions, account balances, and all interfund and all intrafund entries.

GL - PC

- Interactive validation of account numbers from GL/JL when PC transactions are entered.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create/Update GL/JL salary, wage, and fringe benefit budgets.

GL - PY

- Interactive validation of account numbers from GL/JL when PY records are created.
- Budget Checking when PY records are created and payroll is processed.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create new GL/JL Account (Org. Key/Object combination) if previously does not exist.
- PY distribution posts GL/JL transactions and updates account balances with all automatic interfund and intrafund entries.

GL - GM

- Create/Update GL/JL budgets for grant revenue and expenditures.

GL - SI

- Interactive validation of account numbers from GL/JL when SI records are created.
- Budget Checking when SI records are created.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create new GL/JL Account (Org. Key/Object combination) if previously does not exist.
- SI distributions post GL/JL transactions and update account balances with automatic interfund and intrafund entries.

GL - DA

- Interactive validation of account numbers from GL/JL when DA transactions are entered into a set.
- Interactive budget checking when DA transactions are entered into a set.
- Interactive transaction trapping of invalid Org. Key/Object Code combinations.
- Create new GL/JL Account (Org. Key/Object combination) if previously does not exist.
- Distributions post GL/JL transactions and update account balances with automatic interfund and intrafund entries.

GL - NU

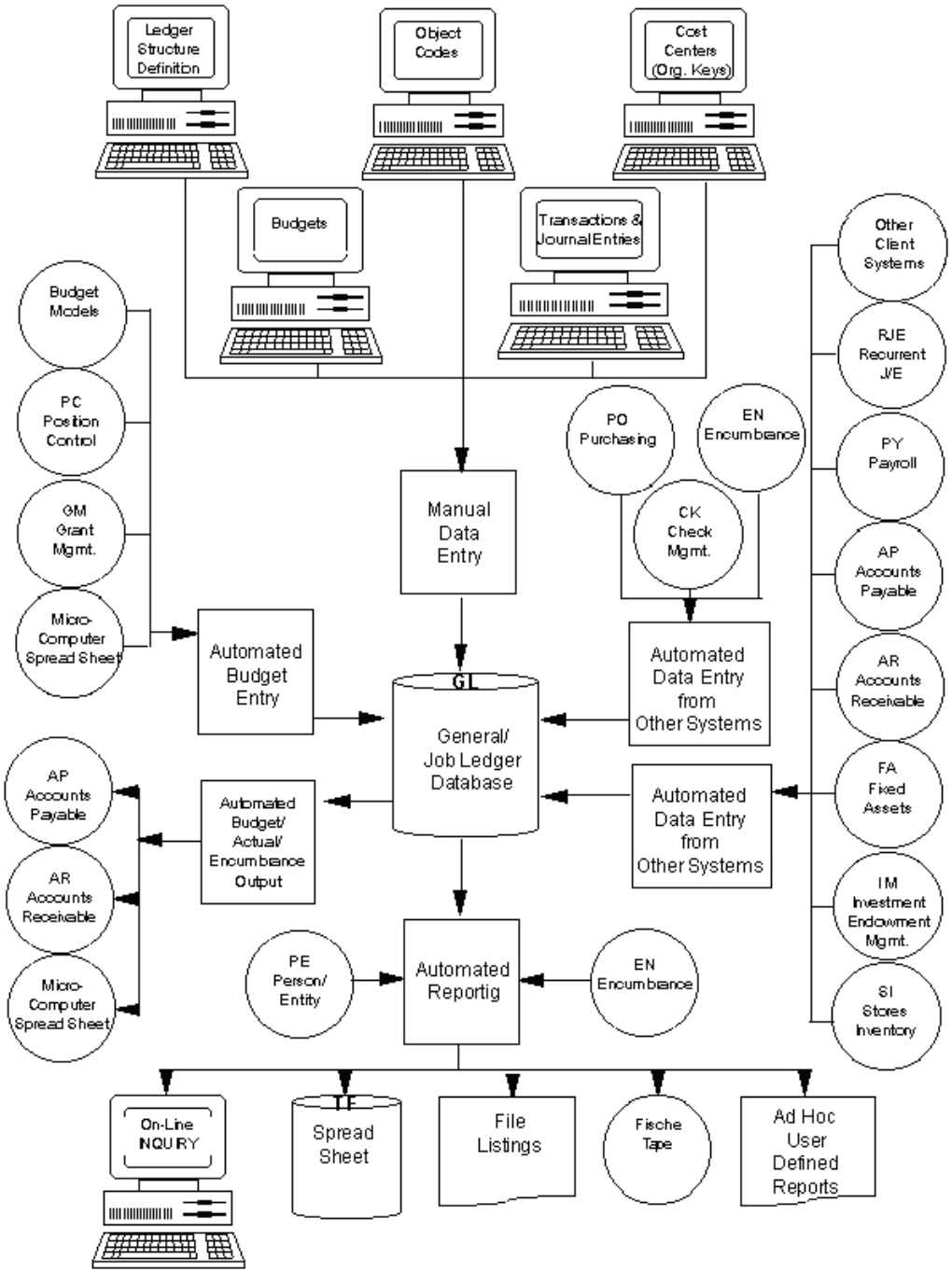
- User security access to accounts and budgets with read, write, or update capabilities.
- User security access to individual menus and job dialogue.
- Audit information for total debit/credit balances.
- Control of open posting periods.
- Maintains all GL posting strategies and preferences.
- Maintains all Bank posting accounts (CKID).
- Maintains Identification codes for each Fund Administration account (FDID).

GL - SY

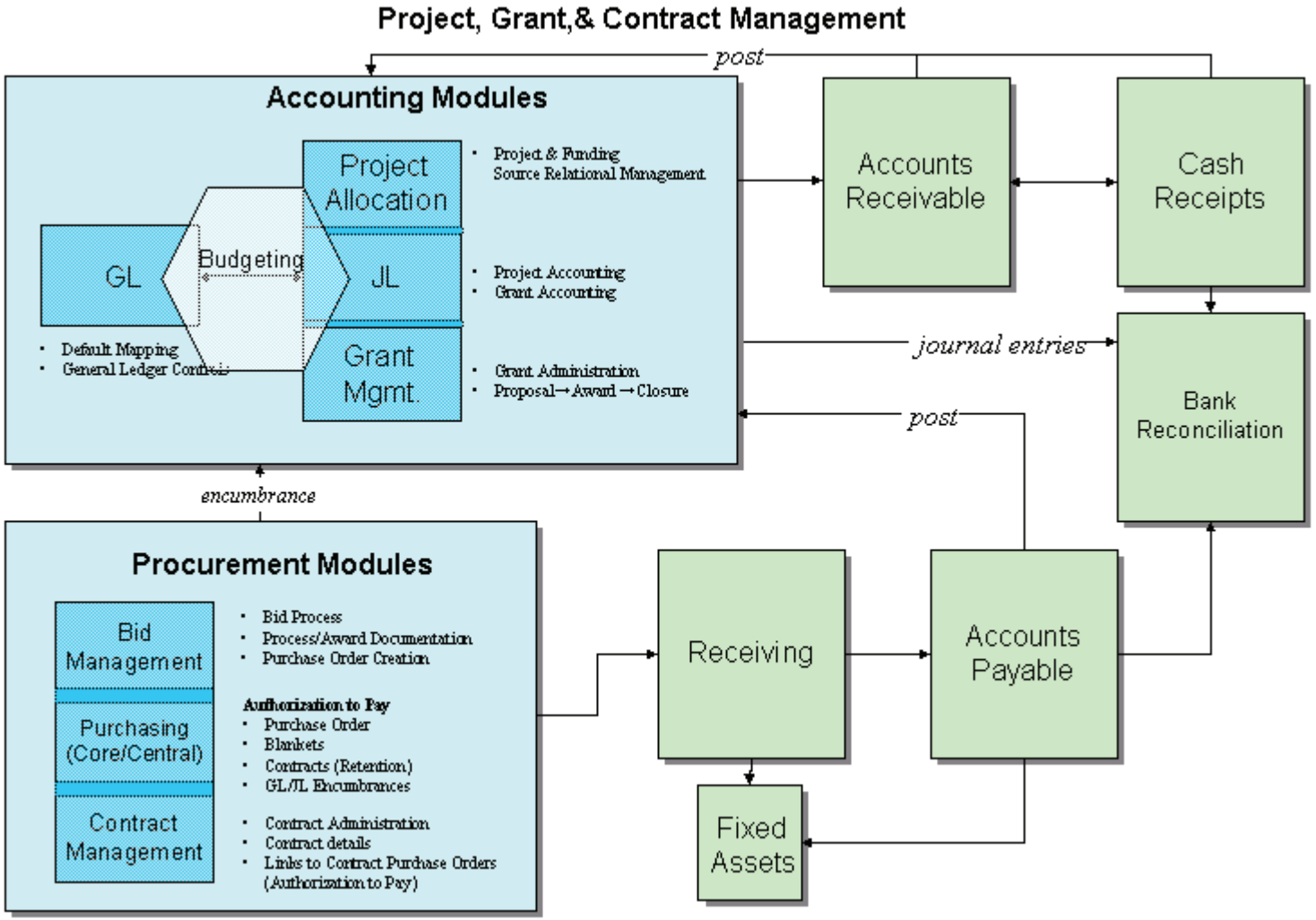
- GLDB re-sizing.
- Adjustment of Report Definition Files.

GL - RX

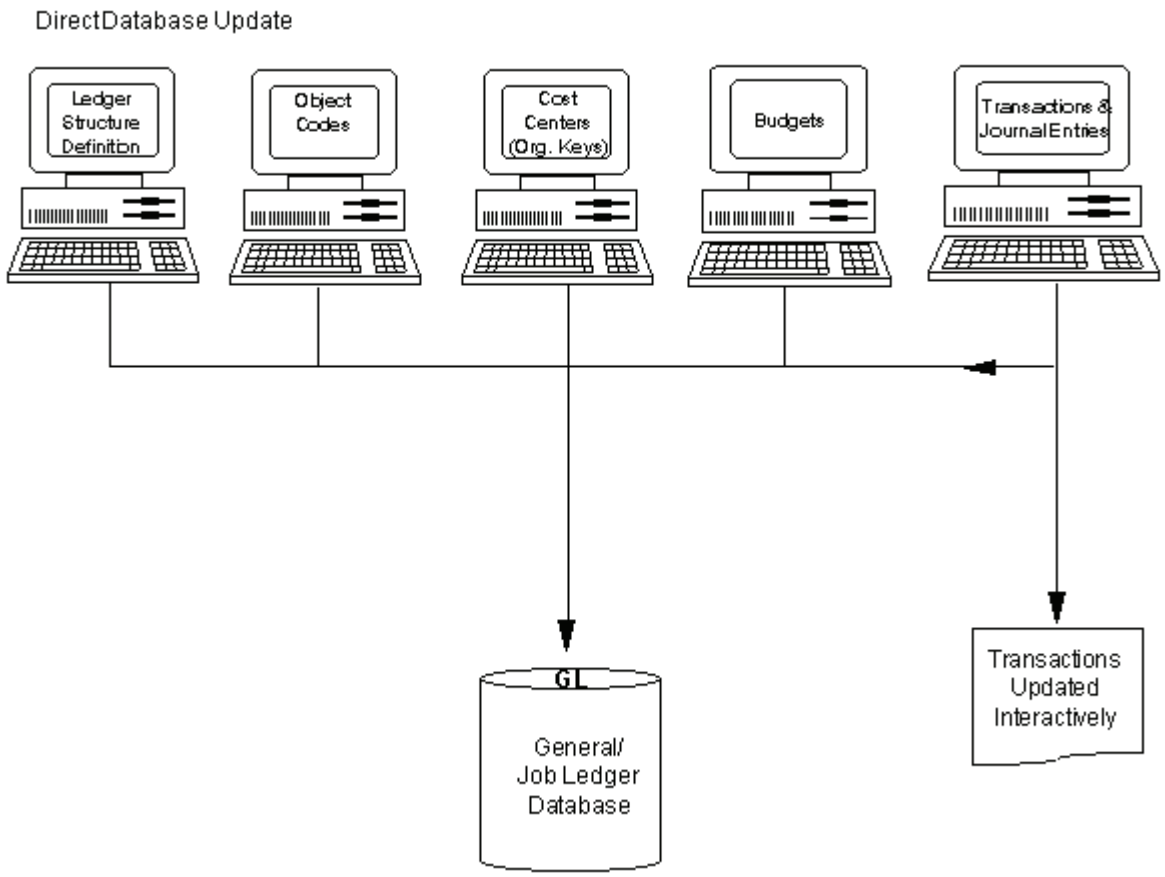
- Report all GL/JL account balances, encumbrances, and budgets for any number of years stored in the GL.


Ledger Structure Definition
Object Codes
Cost Centers (Org. Keys)
Budgets
Transactions & Journal Entries
Budget Models
PC Position Control
GM Grant Mgmt.
Micro-Computer Spread Sheet
Automated Budget Entry
Manual Data Entry
PO Purchasing
EN Encumbance
CK Check Mgmt.
Automated Data Entry from Other Systems
GL
General/ Job Ledger Database
Automated Budget/ Actual/ Encumbance Output
Automated Data Entry from Other Systems
AP Accounts Payable
AR Accounts Receivable
Micro-Computer Spread Sheet
PE Person/ Entity
Automated Reportig
EN Encumbance
Other Client Systems
RJE Recurrent JVE
PY Payroll
AP Accounts Payable
AR Accounts Receivable
FA Fixed Assets
IM Investment Endowment Mgmt.
SI Stores Inventory
On-Line INQUIRY
TF
Spread Sheet
File Listings
Fische Tape
Ad Hoc User Defined Reports

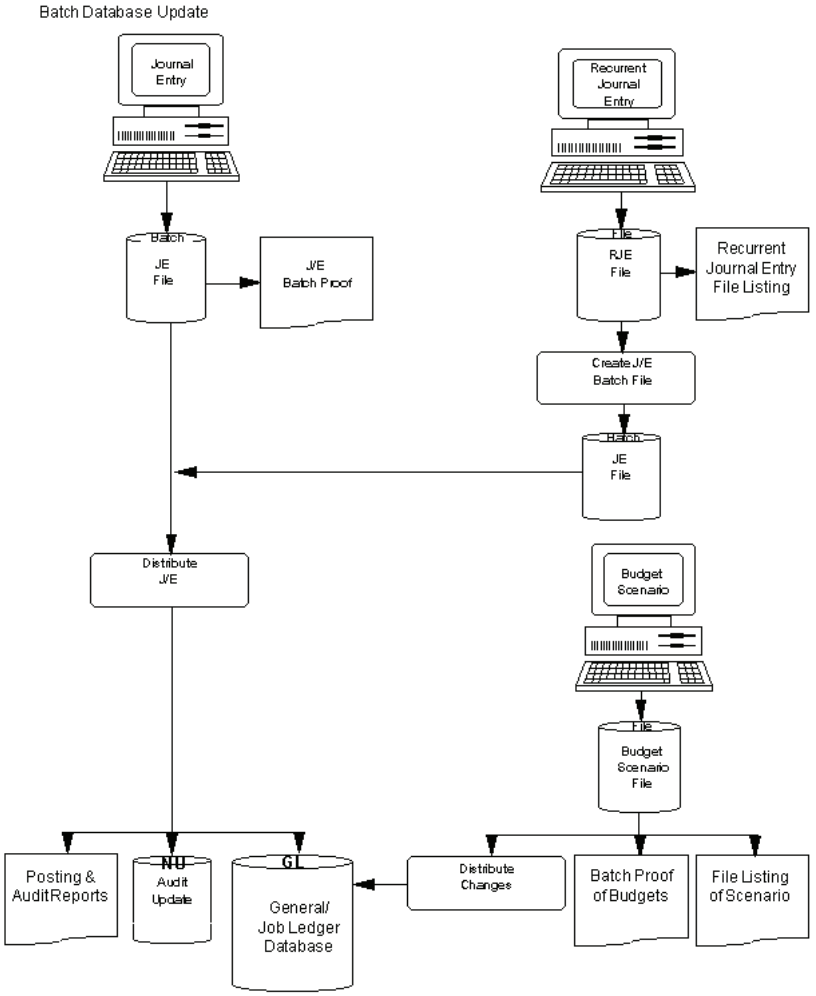
1.2.3 Project, Grant, and Contract Management Integration



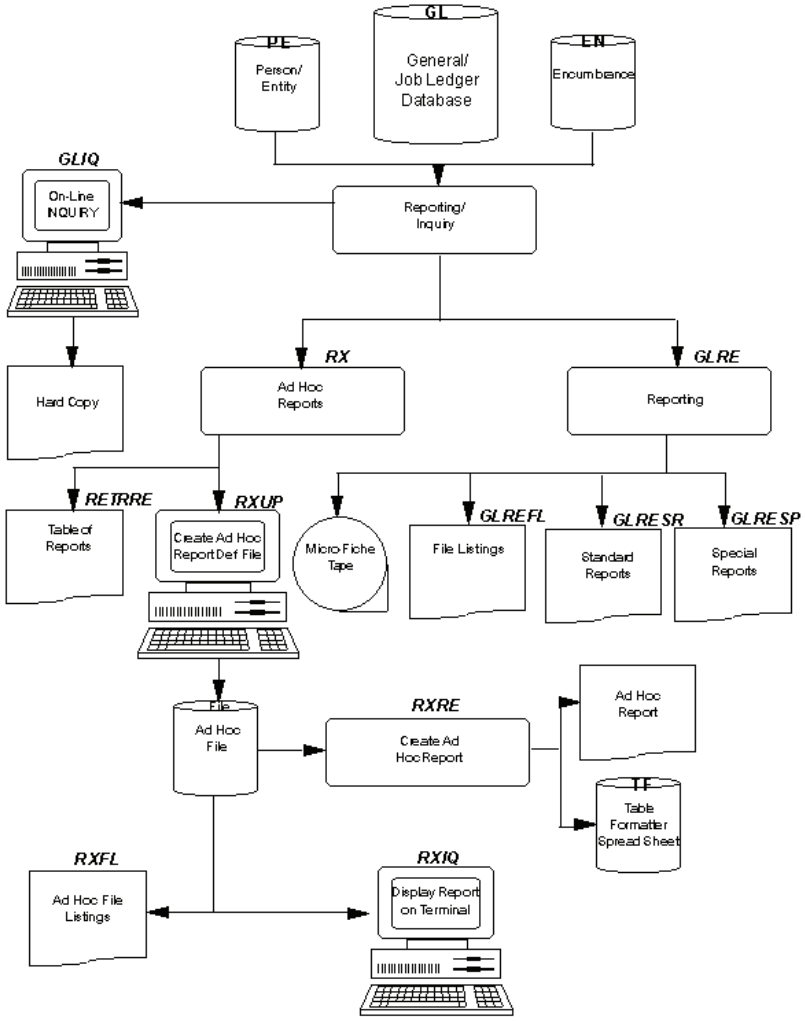
1.2.4 Manual Data Entry



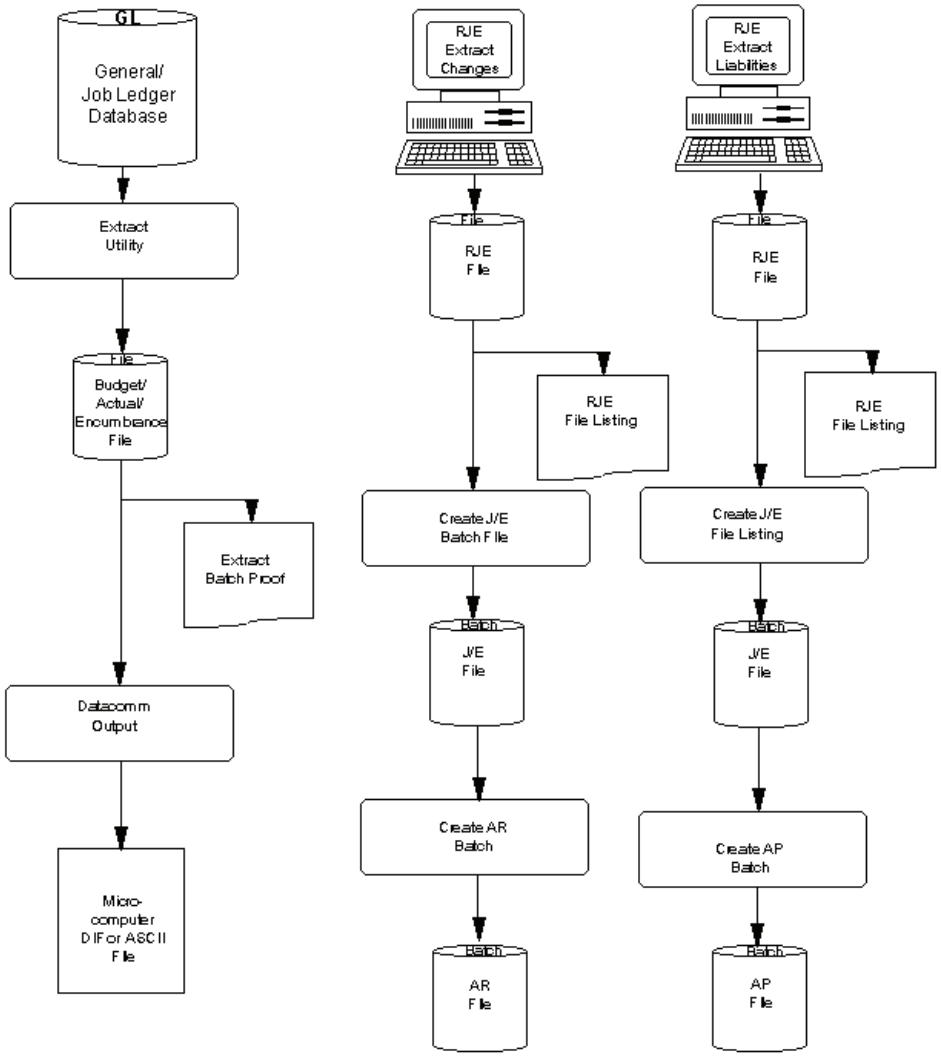
1.2.5 Manual Data Entry 2



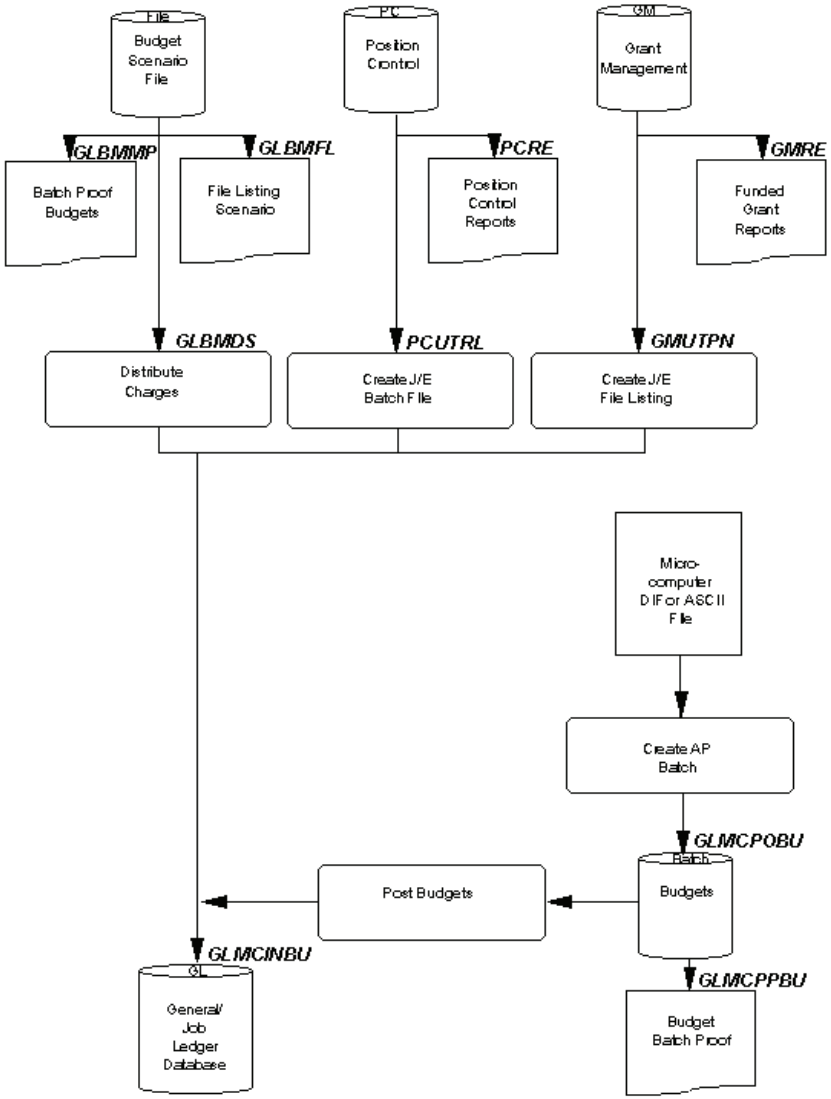
1.2.6 Automated Reporting



1.2.7 Automated Budget/Actual/Encumbrance Output



1.2.8 Automated Budget Entry



1.3 Terminology

GLDB: General Ledger database. The file structures which hold the budgets and transactions that make-up the client's accounting information.

GL Side: One of two parts of the GLDB which contain one or more structural definitions of ledgers.

JL Side: One of two parts of the GLDB which contains one or more structural definitions of ledgers. The JL Side is an optional definition.

Ledger: A formal definition of an accounting structure which is supported in either the GL Side or the JL Side.

Organizational Parts(Org. Parts): discrete part of a Ledger account identification which specifies some aspect of the organization, such as "Fund", "Division", etc.

Org. Key: Numbers or characters which stand for some used combination of values for the Organization Parts.

Object Group: A code which groups like Object Codes into some client defined collection, such as all asset Object Codes or all Salary and Wages Object Codes. Object Groups may be set up for reporting and/or budget controls.

Object Code: A code which represents an entry in a Chart of Accounts. Objects are typically used to classify line-items in an accounting system, but they are also used to classify assets, liabilities, fund balances, and transfers in addition to revenue and expenditures. A unique Object is created for each bank account (assets), for each payroll retirement plan (liabilities), for each type of revenue generated, for each type of expense incurred, for each type of transfer in, and for each type of transfer out. Objects are generally the lowest level of classification, however, some organizations use the term Object Code to represent a grouping of "Sub-Object Codes". This manual will use the term Object to mean the lower level.

Account: The combination of an Org. Key and an Object Code.

Control Key: An Org. Key which has been defined at a higher level within the Chart of Accounts and typically identifies the accumulated sum of budgets for several lower level Org. Keys.

Fully Qualified Account: Input and Output formats which require the entry of all Org. Parts plus the Object Code.

1.4 The General Ledger Structure(s)

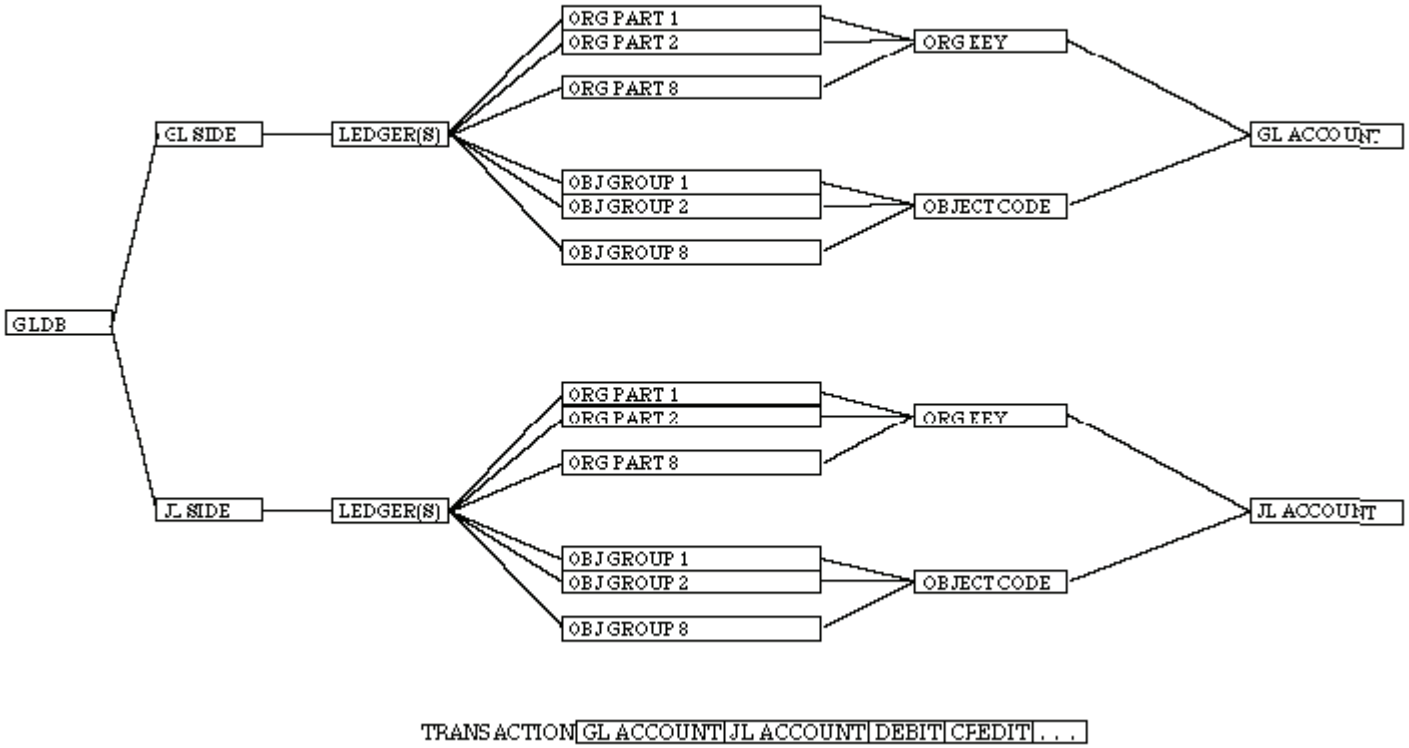
The structural aspects of the General Ledger, as defined by the client, are important because they cast budgets and transactions in a framework for recording and reporting. Budgets and transactions are associated with Accounts. Accounts represent the combination of an Org. Key (the organization part) with an Object Code (the classification of activity). Org. Keys are from one to ten characters in length and may be associated with up to 40 Org. Parts. Object Codes are from one to eight characters in length and may be associated with up to 40 Object Groups. Each Org. Part code and Object Group code may be from one to eight characters in length. Thus, an account which is fully qualified by all of its constituent Org. Parts, Org. Key, Object Groups, and Object Code may be up to 658 characters in length. (8x40 + 10 + 8x40 + 8)

Each transaction held in the General Ledger can specify two Accounts: A primary G/L Account and an optional J/L Account. This is designed to accommodate organizations who wish to do Job Cost or Project accounting. The primary G/L account is generally used to identify the operational side of the organization and the optional J/L account is used to identify the Job Cost or Project side of the organization. The inclusion of two Accounts for each transaction eliminates the need for redundant entry of transactions and guarantees that the G/L and J/L are reconciled to a balanced condition.

As stated in the introduction to the section, this software is used by a wide variety of not-for-profit organizations. The structures used in this guide are intended to portray the common not-for-profit accounting aspects of the system. The client will find that the system will adapt exactly to all specific structural needs of the organization.

This diagram shows the structures as they are cast into the theoretical framework.

NOTE: You can have up to8 primary and 32 secondary parts and groups.



To demonstrate these concepts, we will develop a sample accounting structure. The structure will be used throughout this user guide. The organization described below will be named “The VANILLA Organization", or simply, TVO. The following describes the approach taken by TVO in the defined structures.

1.4.1 GL Side - General Ledger

First, TVO will define its normal operating ledger as the 'GL' Ledger. It represents the standard accounts for all financial activity within the organization. It is named 'GL' since TVO considers this to be their General Operating Ledger for all audited statement activity.

Org. Parts: TVO has four major Org. Parts: A Fund Part, a Function Part, a Division Part, and a Budget Officer Part. Thus, for each used combination of these Org. Part codes, TVO will have one Org. Key. TVO defines each of these Org. Parts as follows:

Fund Part: Two numeric digits, called "FUND"

Function Part: Four alphanumeric characters, called "FNCT"

Division Part: Two alphanumeric characters, called "DIV"

Budget Officer Part: Four alphanumeric characters, called "OFCR"

Org. Key: The Org. Key will be six numeric positions long with the first two positions indicating the Fund code and the last four positions used for uniqueness within a Fund. TVO thinks of an Org. Key as a Department. Thus, they call the Key "DEPT". For each used combination of Org. Part codes, TVO will set up a unique Org. Key.

Object Groups: TVO groups Object Codes with two Object Groups: A Balance Sheet Group and an Income Statement Group. Each Object Group is defined as given below:

Balance Sheet Group: Two numeric digits, called "BAL"

Income Statement Group: Four numeric digits, called "INC"

Object Code: TVO will use four numeric digits to represent the Object Code called "OBJ".

When TVO enters a "GL" Ledger account, they wish to enter the Org. Key, followed by a dash (-), followed by the Object Code. Thus, "101200-6100" would be entered to identify the Accounting Department within Fund 10, and Object Code 6100.

Codes used for the 'GL' Ledger

Ledger	Part	Code Value Meaning
FUND	00	Pooled Cash Fund
FUND	10	General Unrestricted Fund
FUND	20	Capital Improvements
FUND	30	Grants and Contracts
FNCT	N/A	Not Applicable to Function
FNCT	ADMN	General Administration
FNCT	SPSV	Support Services
DIV	NA	Not applicable to Division

DIV	ND	Northern Division
DIV	SD	Southern Division
OFCR	N/A	Not Applicable to Budget Officer
OFCR	VPAD	Vice President Administration
OFCR	VPFA	Vice President Fiscal Affairs

Org. Key	FUND	FNCT	DIV	OFCR	Key Title
000000	00	N/A	NA	N/A	TVO Cash Admin.
100000	10	N/A	NA	N/A	Fund 10 Administration
101200	10	ADMN	ND	VPFA	Accounting Department
101300	10	ADMN	ND	VPFA	Human Resources
200000	20	N/A	NA	N/A	Fund 20 Administration
201200	20	SPSV	ND	VPAD	Maintenance Services
201300	20	SPSV	ND	VPAD	General Services
300000	30	N/A	NA	N/A	Fund 30 Administration
301200	30	ADMN	SD	VPFA	Grant Administration
301300	30	SPSV	SD	VPAD	Daycare in the 1990's

Object Group	Code Value	Meaning
BAL	10	Current Assets
BAL	15	Other Assets
BAL	20	Liabilities
BAL	30	Fund Balances
BAL	40	Revenue
BAL	50	Expenses

INC	1000	Balance Sheet Accounts
INC	2000	Regular Revenue
INC	3000	Special Revenue
INC	4000	Operational Expenses
INC	5000	Salaries and Wages

Object Code	Bal	Inc	Object Description
1021	10	1000	Cash In Bank, A/P Stock ID A
1022	10	1000	Cash in Bank, A/P Stock ID B
1023	10	1000	Cash in Bank, Payroll Stock P
1025	10	1000	Intra-Fund Clearing Account
1029	10	1000	Claim on Cash
1300	10	1000	Accounts Receivable
1400	15	1000	Notes Receivable
1700	10	1000	Inventory Control Account
1800	15	1000	Fixed Assets
1810	15	1000	Accumulated Depreciation
2000	20	1000	Accounts Payable, General
2010	20	1000	Receiving Clearing
2201	20	1000	Employer Fringe Liab. FICA
2202	20	1000	Employer Fringe Liab Benefits
2203	20	1000	Employee Ded. Liab Insurance
2204	20	1000	Employee Ded. Liab Donations
3000	30	1000	Fund Balance
4000	40	2000	Use Fees

4100	40	2000	Activity Fees
4200	40	2000	Funded Grant Revenue
4300	40	3000	Indirect Cost Revenue
4900	40	3000	Discounts Earned
4910	40	3000	Gain or Loss on Sale of Asset
5001	50	5000	Salaries & Wages
5100	50	5000	Employer Fringe Expense, FICA
5110	50	5000	Emp. Fringe Expense Benefits
6000	50	4000	Office Equipment
6100	50	4000	Office Supplies
6200	50	4000	Rent Expense
6205	50	4000	Travel
7900	50	4000	Tax Expense
8900	50	4000	Depreciation Expense

1.4.2 JL Side - Job Ledger (Project Accounting)

TVO would like to be able to track individual Project Activity as it occurs across 'GL' Ledger accounts throughout the organization. To accomplish this, they will begin to use the Job Ledger portion of the General Ledger database. They have chosen to name the Job Ledger 'PR' which stands for Projects.

Org. Parts: TVO has three Org. Parts for Project accounting: A Director Part, a Supervisor Part, and a Location Part. Thus, for each used combination of the Org. Parts, TVO will have one Org. Key. TVO defines each of these Org. Parts as follows:

Director Part: Four alphanumeric characters, called "DIR"

Supervisor Part: Four alphanumeric characters, called "SUPV"

Location Part: Eight alphanumeric characters, called "LOC"

Org. Key: The Org. Key will be three numeric digits long. TVO thinks of an Org. Key as a Project. Thus, they call the Org. Key "PROJ".

Object Groups: TVO will use one very broad Object Group called "BUDG" for Budget Category. It will be one character in length.

Object Code: TVO will use seven characters to represent the Object Code, called "AREA".

Codes Used for the 'PR' Ledger

Org. Part	Code Value	Meaning
DIR	BILL	Bill Jones
DIR	KAY	Kay Smith
SUPV	AMY	Amy Clark
SUPV	FRED	Fred Blake
LOC	SECTIONl	Section one
LOC	SECTION2	Section two

Org. Key	DIR	ENG	MAP	Key Title
200	BILL	FRED	SECTION2	Youth Tennis Classes
300	KAY	FRED	SECTIONl	Spring Celebration
700	KAY	AMY	SECTION1	New Storage Room

Object Group	Code Value	Description
BUDG	E	Equipment
BUDG	L	Labor Costs
BUDG	M	Material Costs
BUDG	R	Revenue
BUDG	S	Set-up Costs

Object Code	BUDG Object	Description
DESIGN	S	Design Costs for Project
ENGIN	S	Engineering Costs for Project

EQUIP	E	Equipment Costs for Project
LABOR	L	Labor Costs for Equipment
MATER	M	Materials Cost for Project
REVENUE	R	Revenue

When the software is installed for TVO, they will be asked to specify how accounts should be entered on data entry screens. For the examples given in this user guide, we will assume that TVO expects to enter accounts with the following conventions:

ffdddd-oooo / jjj-aaaaaaa

where "f" is the Fund, "dddd" is the Department designator, "-" is a separator between the Department and the G/L Object, "oooo" is the G/L Object, " / " is a separator between the "GL" Account and the beginning of the "PR" Account, "jjj" is the Org. Key and "aaaaaaaa" is the Object Code. For example:

"101200-5001 / 200-LABOR"

Represent the Accounting Department within Fund 10, using Object Code 5001, and Project 200 using Object Code "LABOR".

1.4.3 Multiple Ledgers

As time passes, TVO accepts responsibility for the accounting of several independent, external organizations. Since these represent legally separate entities, they do not wish to co-mingle their funds in any way with TVO accounting activity. To accomplish this, they create a separate ledger within the GL Side called 'AG' which stands for Agency Accounts. This separate ledger will be defined with its own unique structure.

Org. Parts: When dealing with Agency accounts, TVO has two major Org. Parts: The Fund Part and the Sponsor Part. TVO defines each of these Org. Parts as follows:

Fund Part: Two numeric digits, called "FUND"

Sponsor Part: Four alphanumeric characters, called "SPON"

Org. Key: The Org. Key will be two numeric digits. TVO thinks of an Org. Key for the Agency Ledger as an Activity. Thus, they call this Org. Key "ACTV".

Object Groups: TVO groups Object Codes with two Object Groups: A Balance Sheet Group and an Income Statement Group. Each Object Group is defined as given below:

Balance Sheet Group: Two numeric digits, called "BAL"

Income Statement Group: Four numeric digits, called "INC"

Object Code: TVO will use four numeric digits to represent the Object Code called "OBJ".

At this point, TVO enters all codes required to define the 'AG' accounting structure outlined above. The steps for this process are exactly the same as for the 'GL' or 'PR' ledgers, except that all codes are associated with the separate ledger named 'AG'.

When TVO enters an 'AG' Ledger account, they wish to enter the Org. Key, followed by a dash (-), followed by the Object Code. Thus, "12-1356" would be entered to identify Activity 12 and Object Code 1356.

1.5 Journal Entries

There are two types of Journal Entries: Standard and Recurrent. Standard Journal Entries specify debit and credit dollar amounts or units which are to be posted directly to the General Ledger. Each distributed amount will be coded to the GL Side and optionally to the JL Side. Standard Journal Entries are placed into a Set File and then distributed to the General Ledger. This seting process permits the separation of data entry from data posting, an important control for most organizations.

There will be very little need for the accountant to prepare Standard Journal Entries. The General Ledger System is fully integrated with all subsidiary ledgers such as Accounts Payable, Accounts Receivable, Payroll, and Fixed Assets. Automatic postings to asset, liability, and fund balance accounts can occur each time transactions are posted to the General Ledger. Cost Allocation and Indirect Cost distributions are performed automatically through the use of Recurrent Journal Entries. Of course, Standard Journal Entries will continue to be used by the accounting staff to make appropriate adjusting entries, reflect intra-organizational charges/payments, special interfund transfers, etc.

The Recurring Calculation module is used to define and create Standard Journal Entries automatically. For example, TVO may post payments to an electric company into some entity level expense account. Then, a Recurring Calculation will be created which spreads the expense to departments based on square footage of office space. Recurrent Journal Entries are typically run each month, and they result in the automatic creation of a Standard Journal Entry Set File. This Standard Journal Entry Set File may then be updated prior to posting to the General Ledger, or the client may elect to have the Standard Journal Entry Set File automatically post when the Recurring Calculation is processed. Other uses of this process include calculating and allocating indirect overhead costs, fringe benefit expense, labor distributions, etc. The system also utilizes the Recurring Calculation module for the preparation of fiscal year end accruals, reversals, closing entries, and balance forward entries. A standard Recurring Calculation is installed with the system for closing and balance forward entries. Clients may customize this to their own unique General Ledger and fiscal year procedures.

An additional feature within IFAS is the ability to record transactions based on foreign currency valuation. All reports may then be generated showing data based on the original currency, U.S. dollars, or converted to other defined currency.

1.6 Reporting and Inquiry

All reporting is done using the IFAS CDD tool. Standard CDD reports are included with the system. New CDD reports can be created or the standard reports can be copied and altered to match your specific needs.

1.7 Budgeting

Budgeting is performed for accounts which generate revenue and incur expenses. Accounts which relate to assets, liabilities and fund balance may also have budgets, although this is not typical. Budgeting may occur on the GL and JL Sides. The system supports monthly, quarterly, and fiscal year budgeting.

The system maintains a maximum of 25 named budget versions. You can reuse the budget versions each fiscal year, with a proposed budget for fiscal year 1 and a different proposed budget for fiscal year 2. Also, for funded grant activity and multi-year projects, the system allows for the entry of multiple year budgets. The user commands the system as to which named version of the budget will act as the current budget. Various budget preparation reports may be generated. The user may wish to create a monthly or quarterly budget, or the user may wish to enter fiscal year budgets on a multi-line form which will permit the entry of several line-item budgets on one screen. For budget version preparation, a special form exists which will display user specified budget and actual information. Then, from this information the user may form a new version of the budget.

Refer to the Budgeting user guide for details about the Budgeting process.

1.8 Relationship to Other Systems (Interfaces)

The General Ledger is the primary ledger of the organization. It is essential that the General Ledger control accounts be reconcilable to other, subsidiary systems, such as Accounts Payable, Accounts Receivable, Cash Receipts, Payroll, and Fixed Assets. The definition of the relationship between these other systems and the General Ledger is a task which is performed during the initial installation visit where SunGard Public Sector IFAS and the client install the GL and JL Ledgers.

Entries are posted to the General Ledger in three levels within the structure: The department/cost center level, the fund level, and the entity level. Generally, the department level has detail transactions posted to revenue and expense accounts, fund level has assets, liabilities and fund balance entries which are held for each fund, and the entity level is generally used for storing cash and investment accounts. The relationship of the department level to the fund level is often represented in an "Intra-fund" Object Code which is associated with the department level Org. Key and the fund level Org. Key. The relationship of the fund level to the entity level is often represented by an "Inter-fund" or "Claim on Cash" Object Code which is associated with the fund level Org. Key and the entity level Org. Key.

Interfaces between other systems and the General Ledger are given by defining, for each type of transaction, which entries are to occur automatically at these three different levels of accounts. This allows all entries to asset, liability (and even fund balance) accounts to be made

automatically by the software. Thus, the accountant need not prepare any such entries at month-end as the entries are made each time transactions are processed.

1.9 Closing and Controls

Two functions are critical to any accounting system. The first is the manner in which it allows control of transaction posting to the G/L by the user and the second is the provision for the fiscal period closing process. When considering the internal controls necessary for any accounting office, it is important to be able to monitor data from five different perspectives: Data entry, authorization, verification, source document maintenance, and historical retrieval.

Data entry should be both consistent and efficient. The transaction creation processes for the General Ledger and each subsystem are always performed in three basic steps: Create/Update a Set File, obtain a Set File Proof Listing, and Post Set File entries. On-line, interactive posting is supported. However, it is nearly always advisable to "Set" transactions so that the data entry function is separated from the data posting function. The security system allows the user to determine not only the timing but also which individuals may perform each step. By separating the creation of Set Files and posting transactions, different levels of authorization and approval are allowed. The accountant may then determine which processes will conform to desired internal control procedures.

Through the extensive security capabilities of the system, complete control of access by individuals to portions of the General Ledger is available from as broad as the Entity level to as specific as the Object level. Also, while certain individuals may be able to create sets of transactions for posting to the General Ledger, they may be easily restricted from all processes which actually cause the posting process to occur. Verification of data entry is provided through extensive reporting to the printer or at the terminal. Each report provides detail, intermediate summary and grand totals for the selected set. Upon completion of every posting process, reports are automatically generated which identify all transaction posting details, verify set totals, and audit total system entries from one series of postings to another.

All data entry processes provide multiple methods by which transactions may be linked to source documents, thereby allowing for a clear audit trail between the system and paper files. For example, the movement of items through Purchasing, Encumbrance, Accounts Payable, Check Management, and General Ledger may be identified by Purchase Request number, Purchase Order number, Invoice Reference, and Check number. Many times, for reference and audit purposes, it is necessary for the user to extract information based on the relationship of items at the time of posting. The system maintains extensive capabilities to extract and report transactions based on the data entry person, specific Set identifiers, and job set entry numbers.

Ease and timeliness are important functions in the closing process. The Accounting Staff is assisted in preparing for the annual financial audit and opening the books for the new fiscal period. Several features within the General Ledger are available which provide an environment for the simultaneous occurrence of ongoing daily processing and fiscal closing activity. First, the system allows posting to any prior or future fiscal period with automatic month-end balance updates. Second, the number of reporting periods within the fiscal year is client defined, up to a maximum of 14. These may be based on calendar months, four week periods, or any other time period. This allows for the separation and identification of unique periods for balance forward entries, audit required adjustments, or any other special closing functions desired by the user. Third, a standard Recurring Calculation file is installed for balance forward and closing entries. It must be adjusted by the client to match

their uniquely defined posting process as required within their General Ledger. These entries may be posted from the individual detail transaction level to any control total level. Also, any subset of the General Ledger may be closed concurrently with other areas or separately, depending on the clients' requirements. Specific accrual and reversal entries may be posted at the same time or separately to the appropriate period. Fourth, the user determines the timing of posting all period closing entries. Special closing entries may be separated from balance forward entries and/or audit required adjustments.

1.10 Installation and Implementation

Installation and implementation of the SunGard Bi-Tech system is a four step process:

1. Establish the client defined structural aspects of the General Ledger and identify the required interfaces to other subsystems.

2. Understanding the various data entry and reporting processes, requirements for internal control, and relationship to existing organizational procedures.

3. Perform adequate testing to insure that the subsystem interfaces and account balancing functions are posting as expected.

4. Enter budgets and initial actual account balances based on the planned "going live" date.

During the initial visit, SunGard Bi-Tech personnel will install the software and assist the client in establishing the structural aspects of the General Ledger. This will include entering Org. Parts and Object Groups into the system to insure it meets the client's requirements. Interfaces between the General Ledger and other subsystems will be defined to conform to the client's specific structure. Basic training will be given which provides a sense for organization of the dialogue menus, data entry processes, reporting capabilities, and system security. Finally, SunGard Bi-Tech will review the special forms which must be customized and assist in determining a conversion schedule. The next step is for the client to exercise the various features of the system. Generally, this will be to complete the entry of all Keys and Objects, test budget data and transactions. Through this, a greater understanding of how to access different portions of the system will be gained. Based on the client's time line, SunGard Bi-Tech will return for detailed training on data entry, transaction posting, and reporting functions. Through this phase, the client will begin to understand how the system may be further customized to blend with existing job responsibilities, organizational procedures, and internal control requirements.

The third step, normally, is to perform detailed testing of data entry and transaction postings. This can be as limited as matching several sets of transactions from the old and new systems or may entail "running parallel" with the existing accounting system for a specified period of time. The important function is to determine that all subsystem interfaces and self balancing features are performing as expected. The fourth step is to prepare the system for "going live". This requires the entry of approved budgets for all accounts and entering initial transaction actual balances. Depending on the client's needs and start-up plans, SunGard Bi-Tech will assist in "rolling" this information from the old system to the General Ledger.

1.11 Set up system parameters

GL System parameters are defined using the GL General Setup screen (GLUPGN). There you define information which tell IFAS how the system is to work with the General Ledger. For example, how many periods are in a fiscal year, which month is the first month of the fiscal year, what names the user wants to use for various versions of budgets which may be prepared, what the organizational parts, such as Fund, Function, Division, Budget Officer, Cost Center, etc. will be called, what the object groups, such as Budget Category, Income Statement Groups, Balance Sheet Groups, etc. will be called, and what the period calendar looks like.

1.11.1 Define Organizational Part Codes

In step 1 the system parameters were defined. Part of those system parameters tell IFAS what the parts of the given Ledger should be called. This step is where the system is told which codes are valid for the various Organizational Parts. For example, if the General Ledger has a Fund part, the user will now call up a screen on which will be defined the Fund codes and associated descriptions. These codes need to be set up so that the system can validate them when they are used to set up Org. Keys within the organization. These parts are also useful for report selection criteria, sorting, and report development. To access the screen that allows definition of the Organizational Part codes, key in **GLUPKY**.

1.11.2 Define Subsystem Interfaces

When transactions are processed by the various subsystems, such as Accounts Payable, Accounts Receivable, Payroll, Cash Receipts, and Fixed Assets, there are postings which need to be made to the General Ledger database. IFAS is extremely flexible in terms of the posting strategies which may be employed in these automatic entries. This flexibility means that postings will happen exactly the way the user wants them to, and it means that someone needs to learn how to tell the system how they should happen. The system is installed with some fairly standard strategies for automatic posting. The accountants then make changes to or add strategies. These postings are defined to the system so that there is no need for manually created Journal Entries to post to accounts such as "Cash in Bank", "Accounts Receivable" or "Accounts Payable" (i.e., we would like the system to make these entries automatically, each time transactions are processed). The general method for defining these subsystem interfaces (automatic entries to the General Ledger) is to call up the screen (on which each subsystem and type of transaction is specified) and define the types of entries to be made to the General Ledger. To access the screen which allows definition of the subsystem interfaces, key in **GLUTSPSI**.

1.11.3 Learn How to Use the Security System

IFAS includes a comprehensive security system which will allow the user to specify who can update, read, write and/or delete what information and when. For example, department heads may only need to review budgets which relate to their responsibility area. A

comprehensive review of the security manual will provide information on how to define users with different types of security. Signing on as those users can further demonstrate the capabilities available with each of the different types of security.

2 The General Ledger Structure

2.1 General Information

The following sections of this User Guide are prepared in a manner such that each form is described in general terms in the guide along with examples and helpful hints of various features. Of course, the user may also review each form and access the on-line help facility at any time.

The software is designed to be extremely flexible so that it can accommodate every client's needs. In order to accomplish this, the client must specify how the software is to operate. This is done for the General Ledger on the **GLUPGN** screen. The first six general information forms must be completed by the user as the first formal step in the creation of the accounting system. The seventh form is optional. The information which follows will instruct the user on how to complete each of these six forms.

Note to California clients: SACS compliance requires consideration during the initial planning stages of your accounting structure. If SACS compliance is required for your organization, refer to the California section of the State Regulatory Reporting guide.

2.1.1 Organization Parts Definition

This is used to tell what Organization Parts exist for the given Ledger Code. The Organization Parts are background codes which represent formal structural definitions of the Ledger. Each Organization Part should define logical, discrete areas of the organization or financial structure. These might relate to different hierarchical levels within the organization which require reports grouped or summarized for their area of responsibility. Examples might be executive officers, mid-level administrative departments, Deans, or Academic Divisions. From an external financial reporting perspective, Org. Parts typically identify Fund Groups, Funds, Sub-funds and Functional or Categorical areas.

Two methods of account definition exist within IFAS, one is the Organization Key the other is the Fully Qualified Account. Each will be described separately within this section.

Organization Key

Recall that there are two types of Ledgers. General Ledgers, which have a timeline involving fiscal years and can stand on their own, and Job Ledgers, which must be associated with a General Ledger and can have a timeline that is either a fiscal year or from the Inception of the Job Ledger. Every GL or JL is made up of Org. Keys which describe "Where" the money is involved or "Who" it was involved with, not "What" it was involved with. **For example a "Where" could include the General Fund, for Safety, Police Division, and K9 unit. It does not include "What" like dog food or gas.** The Org. Key represents a set of Org. Parts. In our example for the GL ledger, the Org. Key is called a Key. Each Key will be associated with an appropriate Fund, Function, Division, and Officer code from the defined list of Org. Parts. The length of the Org. Key is six characters and will always be numeric digits. At data entry, the user wishes to use the following input format:

Key - Object

This form is used to define the length, type, and description of the Org. Key. The maximum length is ten characters or digits for the Org. Key and Org. Parts. For the Org. Key and each associated Organization Part code, the user will provide a short, medium, and long description. These are used by the software to automatically tailor all screens, error messages, prompts, dialogue, and reports to use the client's terminology and structure. This screen must be filled in for each Ledger Code defined by the user. Since the software automatically writes this information when a Ledger Code is first entered, these values may only be set or amended in BROWSE-mode.

The input and output formats are echoed from the Presentation, Form 5. If they are blank, it means that they have not yet been defined for this ledger.

The Fund, Function, and Budget Officer radio buttons are used by the system to automatically generate various standard reports which summarize data based on these typical Fund Accounting Groups. After defining all Org. Parts which will be used in a ledger, indicate by Part Number which ones should be used in grouping these three levels of reporting.

The Secondary Key Parts Tab allows the user to enter up to 32 additional key parts. Unlike the Primary Key Parts which are always required if they are a part of the framework to be entered when creating a new key, the user can choose whether or not the Secondary key parts are required. The Secondary Key Parts can be used in a way similar to the Primary Key Parts as a functional part of the Key, or they can be used only for reporting purposes like the Selection Codes. An example for the use of an additional key part is if there is a special report that the Mayor wants to see every two weeks for the city council meeting. This report is to include tax revenue, parks and recreation spending, sewer spending, and street sweeping, a special secondary part could be made so that all of these items could quickly and easily be shown in a report since these park links all of these areas together.

The Selection Code Description Fields are optional and may be very useful when performing special studies or analyses. While they are not a part of the formal General Ledger structure, they can be used to define additional levels of summary or grouping for Org. Keys. The client is able to define the heading or title which is to be used for one or more of the maximum eight fields. Each entered title will appear on individual Org. Part Key Definition forms and unique codes may be entered such that each title can be extracted or used in various reports. See the section on defining Organization Key Codes. Selection Codes are used rarely in favor of the new Secondary Key Parts

2.1.2 Object Structure

Once again the Database is made up of General Ledgers and or Job Ledgers. These are made up of Org. Keys and Objects. Keys describe the “Where and Who” of a transaction, and Objects describe the “What” of a transaction. Objects have background GROUPS for organization, reporting, and functional purposes. Typical Groups include summary levels to produce the Balance Sheet, Income Statement, or Budget Categories for budgeting at levels above the Object Code. The first of these groups should always be Type and include: Assets, Liabilities, Equity (Fund Balance), Revenues, Expenses, Transfers In, and Transfers Out. **For Example: an Object could be Pencils. This could have Groups such as (TYPE) Expense, and (DETAIL TYPE) Supplies.** This is the second form and it is used to tell the software which client defined GROUPS are to be used to combine Object Codes. As discussed in the previous form, these Object Groups are background codes

which represent formal parts of the Ledger to which an Object Code relates. Object Groups should define logical summary levels needed for budgeting and/or reporting.

As will be discussed later, Object Groups may be defined such that budgets are created and budget checking occurs at any level within the structure. If this is a desired feature of the system, those Object Groups which will be used as a basis for budgets **MUST** be defined in a strict hierarchical form for budget entry, checking, and subsequent reporting. For example, we might define three (3) different Obj. Groups which will be used in budgeting; Budget Category, Budget Area, and Budget Group. They are defined in a hierarchy such that Budget Category is subordinate to Budget Area which is subordinate to Budget Group. Each Object Code will be associated with these three levels. Object Codes defined within a Budget Category cannot be independently defined to separate Budget Areas and Budget Groups. The following chart may help to see this graphically.

This form defines the length, type, and description of the Object Code. The maximum length is eight characters or digits for the Object Code and eight characters or digits for each of the Object Groups. For the Object Code and each associated Object Group, provide a short, medium and long description. These are used by the software to automatically tailor all screens, error messages, prompts, dialogue and reports to use the client's terminology and structure. This screen must be filled in for each Ledger Code defined by the user. Since the software automatically writes this information when a Ledger Code is first entered, these values may only be set or amended in BROWSE-mode. In the example of the GL Ledger, "Object" will be the description used.

The Balance Sheet and Income Statement Grouping radio buttons have no practical functionality currently.

As mentioned previously, when using the 'Padding' feature, it may be helpful from a daily data entry perspective to have all padding be applied consistently; thereby increasing data entry speed and individual sight recognition.

The Object Code Subsystem Field Titles identify which subsystems may post to Object Codes within a defined Ledger. When defining individual Object Codes, the software will also provide the opportunity to indicate, of the allowable subsystems, which ones may post to a specific Object Code. (See the section on Object Code Definition).

Like on the Org Key Screen, there is a Secondary Tab for Object Groups. This will allow the user to enter up to 32 extra Object Groups which can be required to enter a new object or Optional.

2.1.3 Calendar Definition

This is the third form and it is used to specify the calendar which exists for each defined Ledger. On the Calendar Definition form the Name, Start Date, and End Date which will be used by the system to identify report headings, posting date periods, and period reports are entered. For the example, 12 periods have been defined, beginning with July and ending in the following June with an extra period defined for closing entries. Each period is one calendar month indicated by the MM/YY format for Start and End Dates.

2.1.4 Budget Version Definition

The fourth form is used to specify up to 25 named versions of the budget permitted for each defined Ledger. For each named version, enter a two-character version code, medium and long description, a date code, and Y/N if budget changes should/shouldn't be logged. The date code indicates the last date upon which revisions to budgets for a version may be entered or updated. Presently, supported Date Codes are the month and year entered in the MM/DD/YY format. The year may take the form of '90' for a specific year, 'FY' to indicate same month and day for each fiscal year, 'FY-1' or 'FY+1' to indicate last or next fiscal year respectively. For instance, the 'GL' example shows named budget versions of PR, RE, AP, DA etc; The approved budget 'AP' is always open and budget changes will be logged. Additionally, no logging of budget changes will be made on versions 'PR and RE'. Note that swapping budget versions is not allowed because it leads to inappropriate results.

It is possible to have a budget that is a combination of other budgets. This is called a derived budget. A derived budget cannot be changed, except by changing the budgets from which it is derived from. **The 'CB' Current Budget is an example of a derived budget.**

Note: It is possible to subtract one budget from another by using negative signs instead of positive signs (each budget listed in the date code takes on the closest sign to its left unless there is no sign, in which case it is considered to be positive). It is best to avoid negative totals if you are using monthly budgets. The monthly budgets for a summed budget are displayed as percentages. Since a negative percentage makes no sense, negative monthly totals are zeroed out and the amount is divided among the other months.

2.1.5 Presentation

Account input formats, used at data entry time, may be in the form of the Org. Key - Object or Fully Qualified Account methods. The procedure for definition is outlined below. Section II.C.1 also discusses use of the Quick Account Number for additional, shortened account data entry. IFAS utilizes a hierarchical approach in determining which account method has actually been input to data entry screens. First, the system checks to see the defined input format on the form shown below. If it is a Fully Qualified Account, IFAS uses the following logic for steps one through seven. If it is the Org. Key - Object method, IFAS uses the following logic for steps four through seven:

1. Has the account been entered as a Fully Qualified Account?
2. Have all Org. Parts plus the Object been entered?
3. Are the entered codes valid?

If Not

4. Has the account been entered as Org. Key - Object?
5. Are the entered codes valid?

If Not

6. Has the Quick Account Number been entered?
7. Is the code valid?

As can be seen, even though the Fully Qualified Account method may be chosen, different account data entry options are available depending on the users' desired method of entry.

The codes used to determine the input and output formats are:

'K' = Org. Key, 'O' = Object Code,

'P1, P2, P3 . . . P8' = Defined Org. Parts

'G1, G2, G3 . . . G8' = Defined Obj. Groups.

To define the method which will be used for input and output, indicate the order in which each part of the Ledger should be presented. For instance, in our example, we would like to input, at data entry time, the Org. Key and Object Code with no separation. This is defined as; 'KO'. 'K' indicates that the Org. Key is entered first and all six digits are entered. 'O' indicates that the Object is entered second without a space or character separating it from the key and all four digits are entered. The input format cannot have spaces in IT. For example, "KO" is OK, but “K O” is not. Hyphens with the codes K, O, P1, etc. can be used in the input and the output formats. Other characters (such as periods) should NOT be used. For the output of report headings, we want the Org. Key and Object Code separated by a hyphen. This is defined as; 'K-O'. Another valid definition for the output format would be to separate the Org. Key into two parts and include the Function and Income Statement codes. This is described as; 'P2-K3-O4-K3-G2'. 'P2' says Org. Part #2 appears first. The first three (3) digits of the Org. Key are second. All four (4) digits of the Object Code are third, followed by the last three (3) digits of the Org. Key. 'G2' says Object Group #2 appears last. At this time only the Org. Key may be separated into two parts. The user may also indicate the description to appear above the output format on all reports.

If the output account number is twenty (20) characters or greater, the system will reserve a maximum space of forty (40) characters for printing on all reports. In all other circumstances, the space will be nineteen (19) characters for printing.

Once this information is defined, it is echoed on the Org. Key and Object Code definition forms.

2.1.6 Miscellaneous

This sheet contains various checkboxes and input boxes that control various aspects of the General Ledger. This sheet is explained in the online help however there are several things to note.

This sheet is where you associate a General Ledger with up to 10 Job Ledgers. Associating a GL with a JL involves having the user, while in this sheet under the General Ledger, type the 2 letter code for the JL in one of the 10 "Valid JL Association Boxes." This needs to be done on this sheet before performing other functions that involve the JL.

The Intra Fund/ Inter Fund Object code boxes are important during the posting phase of the GL. This will be talked about more in the GL Subsystems Chapter.

After this Screen has had the appropriate changes made as well as the other GLUPGN Screens, if ENTER is pressed the computer will tell the user that a record has been added. The user has just made the framework of a ledger.

2.1.7 Associated Text

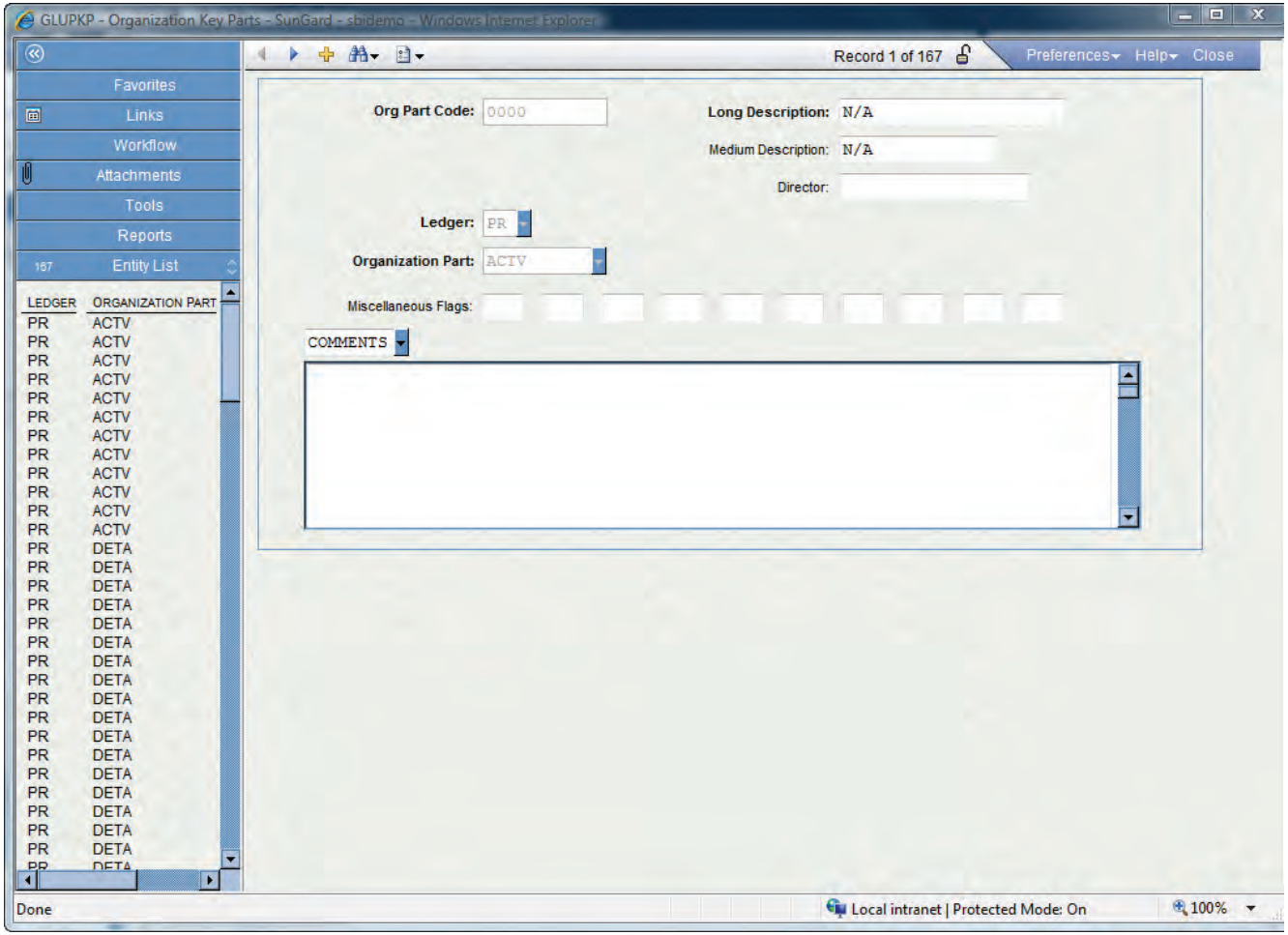
The seventh form allows optional free form text to be entered which relates to a specific Ledger Code. The text entered on this screen may optionally be printed on the GLREFLGN file listing report. In the Ledger field, enter a valid GL or JL side ledger code to associate text with. In the Line column, put a line number. In the text column on the same line, enter text describing the Ledger shown at the top of the screen. An unlimited number of lines may be entered in this manner.

2.2 Organization Keys and Object Codes

There are different ways to define and update Org. Keys and Object Codes. During initial installation it is usually helpful when entering large numbers of new codes to the system to access both Org. Keys and Object Codes at the same time. This may be achieved by accessing either the Organization Key Definition (**GLUPKY**) or Object Definition forms (**GLUPOB**). GLUPKY is used to create keys using the framework made in GLUPGN on the Key tab and is linked to GLUPKP which is the form to define the background parts for the Org. Keys. GLUPOB is used to create objects using the framework made in GLUPGN on the Object Tab and is linked to GLUPOG which is the form to define the background parts for Object Groups.

2.2.1 Organization Key Parts Codes

For each Organization Part associated with a given Ledger definition, this form is used to specify the valid code values which may be given to a specific Organization Part. All Org. Key Parts Codes must be defined before defining Org. Keys. Each value entered has a long and short description and is associated with a previously defined Organization Part which was defined on the GLUPGN. A Ledger Code must be indicated since the system will automatically check to see that the Organization Part has been defined. The Organization Part Miscellaneous Flags are not functional at this time.



2.2.2 Organization Key Definition

After all Organization Key Parts Codes have been defined, go to the GLUPKY form. This is the form on which each Org. Key is defined. Each Org. Key will be used to represent a valid combination of the respective Organization Parts for a given Ledger. Flags and codes are also indicated which determine the transaction traps and budget checking which are desired. To learn more about budgeting see the budgeting guide.

In some instances, it may be appropriate for Org. Keys to be defined as a higher level or Control Key. They represent some portion of the Chart of Accounts to which transaction activity is **not** recorded but rather act as budget control or for other information purposes. When defining Control Keys used for budget values only, the **Key Type** field should indicate 'C' and the **Accept Tr?** field should indicate 'N'. This insures that the Org. Key will NOT be included within regular transactional reporting and will NOT have transaction activity entered against it. In addition, the budgets for a Control Key may be derived from other lower level Org. Keys. That is, its budget values are the accumulated sum of the budgets which have been entered for other Org. Keys. For this to occur, two items must be defined. One, the Control Key must have a 'Y' in the **Derived** field. Two, all appropriate lower level Org. Keys must have the Control Key value entered in the **Budget Checking** field.

In other situations, an Org. Key may receive its budgets as an allocation from a higher level Control Key. That is, the budgets were entered for a Control Key and the user wishes to insure that when entering budgets for a specific Org. Key that it does not exceed the accumulated total of all other budgets which have been previously entered in other Org. Keys which are also allocated from the Control Key. This is accomplished by entering the Control Key value in the **Budget Checking** field and placing one of the following codes in the **Allocation** field:

WO Warning at the Object Level

BO Block at the Object Level

W1, W2 . . . W8 Warning at an Object Group Level

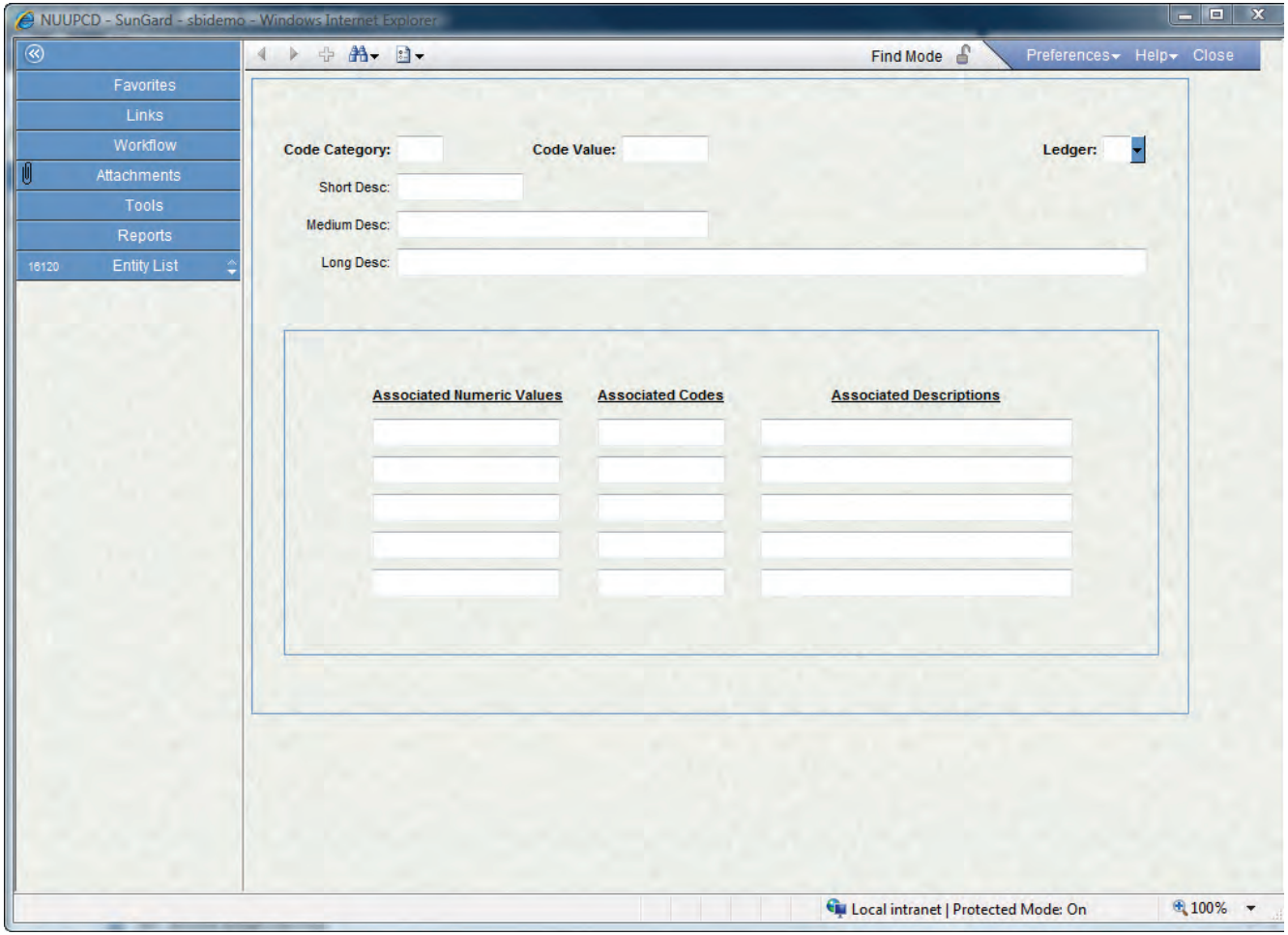
B1, B2 . . . B8 Block at an Object Group Level

The system assumes that budget checking will be at the Object Code level. The **Budget** field allows the user to alter this and also indicate the type of budget control which should occur for all Object Codes associated with the Org. Key. The Budget field is made up of two separate characters. The first indicates the type of control which the system should exercise. If situations at data entry exist where a budget overrun is to be blocked before allowing transactions, then enter a 'B'. However, if only a warning is to be given enter a 'W'. If the user does not desire budget checking, leave the first character blank. The second character indicates at what level the checking is to occur. Valid codes are 'O' (default) for Object or '1-8' for the Object Group Parts.

Each of the defined background Organization Key Parts are echoed on this form and must have an appropriate Org. Part Code entered. Once all Org. Part Codes have been associated with an Org. Key and the 'ENTER' key is pressed, the descriptions for each code will appear.

If Select Descriptions were defined on the ORGANIZATION STRUCTURE form through **GLUPGN**, then those descriptions will appear above the eight (8) fields on the lower portion of the form. Each individual Select Code for a given description must be defined within

NUCLEUS Common Codes, **NUUPCD**, before it may be entered. The Code Category is defined in the form of "xxSi", where "xx" is the two character Ledger code (i.e., GL) followed by "Si", where "i" is the field position of the Select Description (i.e., 'S1', 'S2' . . . 'S8'). The Code Value is the actual Select Code which should be validated. The Medium Description will be used in reporting. All other Common Code fields are not required. The following example shows a Select Code for the GL Ledger for Select Description number one and the value to be validated is 'FDGRNT'.



Optionally, eight (8) Miscellaneous Code fields are provided for additional grouping. The codes entered are not validated and may be entered at any time.

In some instances it is desirable to control the types of Object Codes which may be associated with an Org. Key. By placing a 'Y' in the "**Require Association?"** field, the Object Codes must have been previously associated with the Org. Key before they will be allowed at data

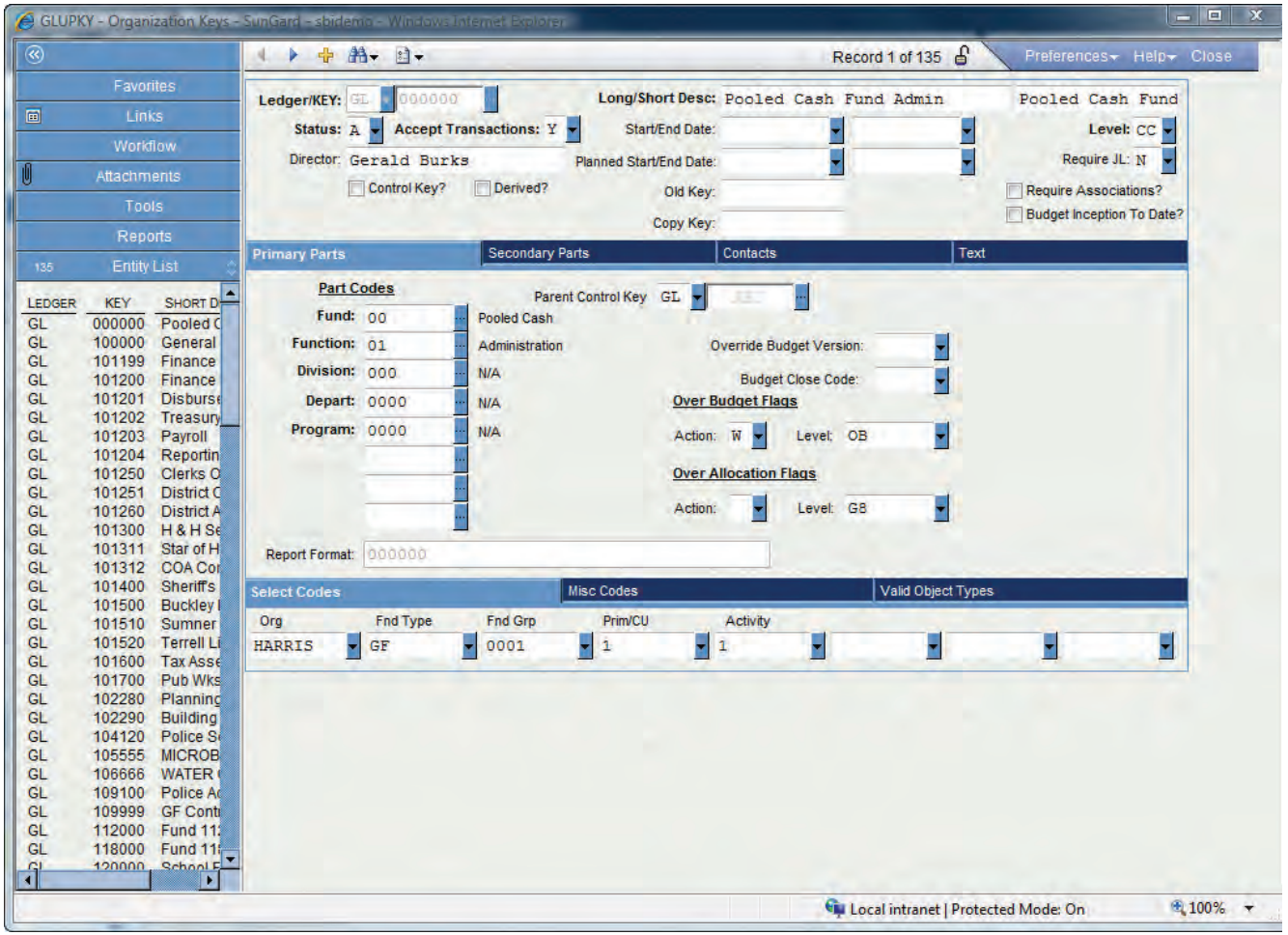
entry. This association may take place in two ways: one, by entering transactions before setting the 'Y', or two, by entering a budget. Additionally, the user may indicate up to six (6) valid Object Code Types which may be associated with an Org. Key. For instance, the user wants Org. Key 101200 to only have revenue and expense objects. Therefore, the type codes of 'RV' and 'XP' would be entered.

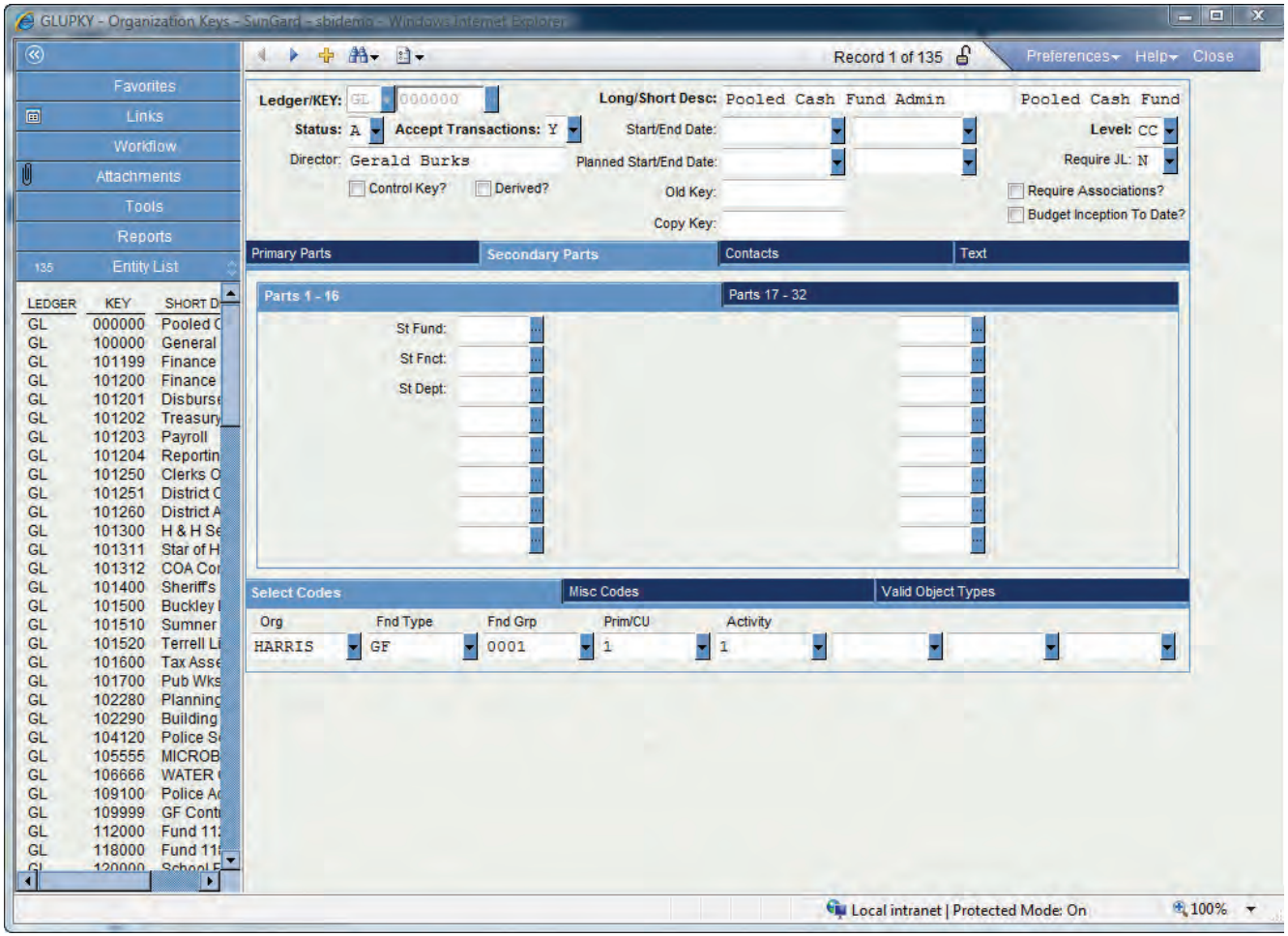
Even though a 'Named Version' of the budget may have been indicated as the default version for a particular 'side', each Org. Key may override the default and have another 'Named Version' of the budget used for budget checking and reporting purposes. Also, under certain circumstances it may be desirable to require a JL Account when a specific Org. Key is entered at data entry time. This can be accomplished by placing a 'Y' in the "**Require JL?"** field.

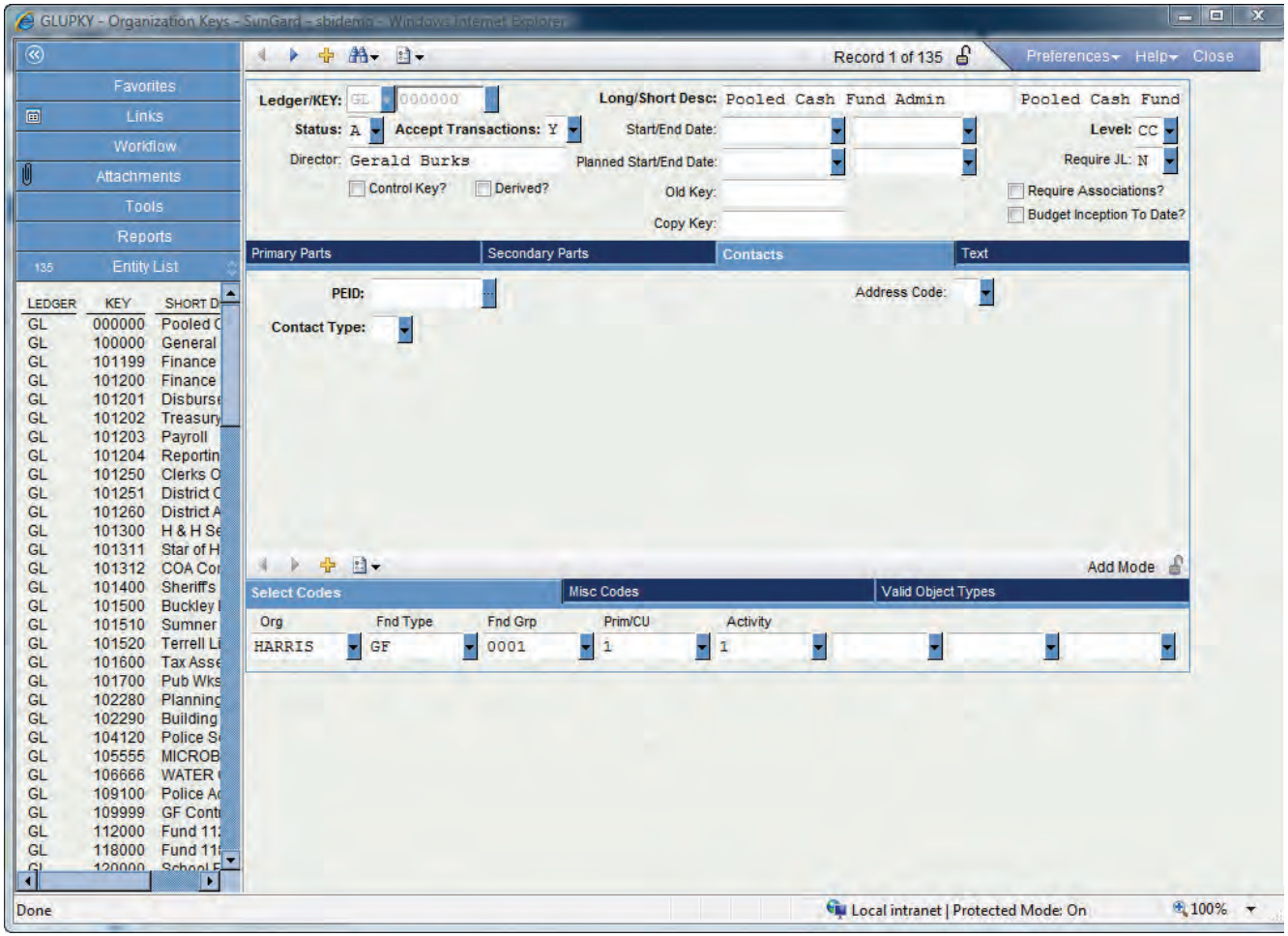
When establishing large numbers of new Org. Keys, the **Copy Key** field allows the user to copy all attributes of an Org. Key to the new Key being defined. Also, at the time of installation or conversion to IFAS the old account number from the previous system may be indicated in the **Old Key** field. This allows a clear historical audit trail for rollover reporting and eases the mapping of old to new accounts for the transaction or balance rollover.

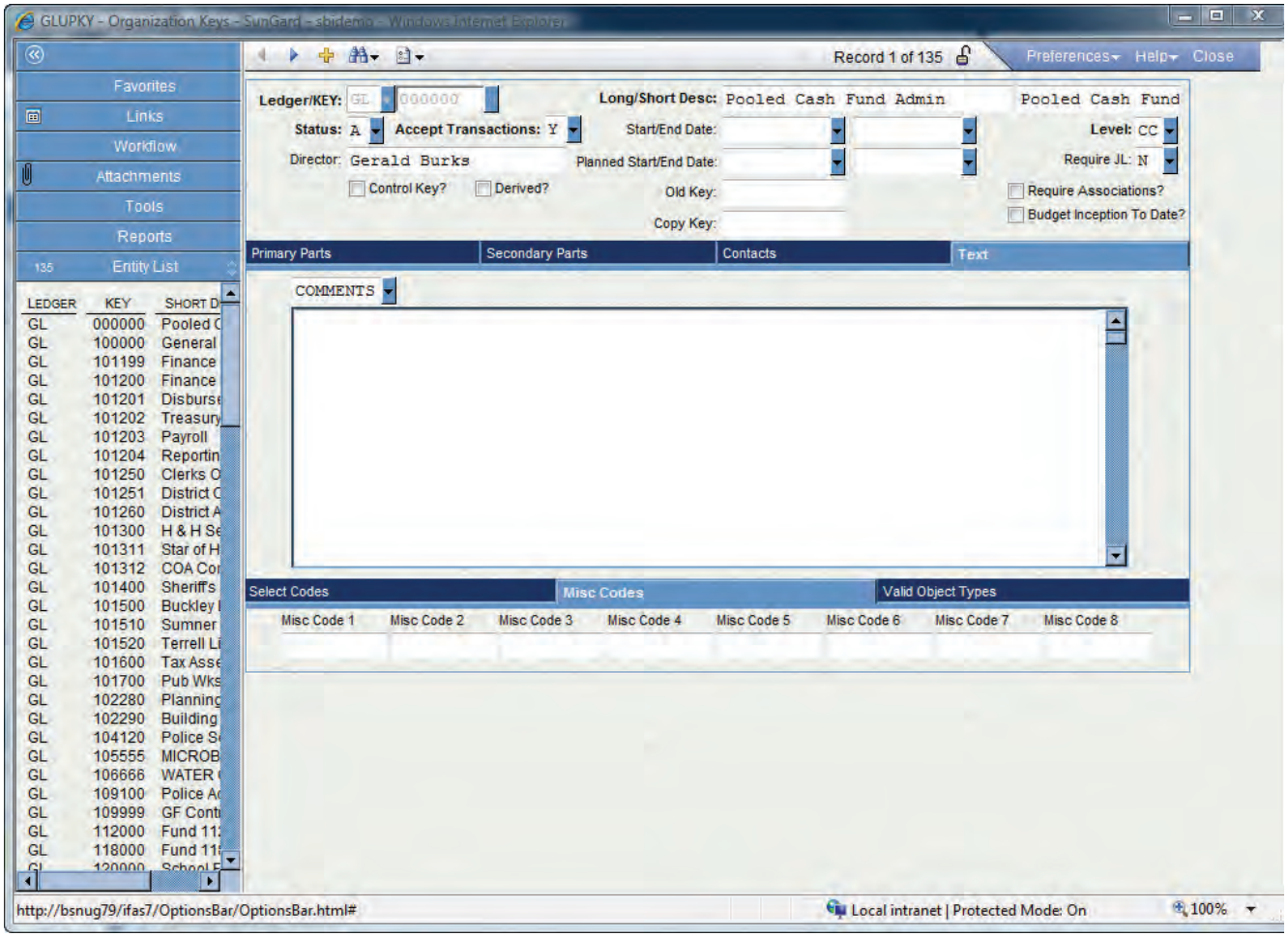
The ledger code in "xxSi" is GL or JL. The specific financial ledger is specified in "Ledger: __ ". The two "Ledgers" are not the same code. If a select code is valid for several GL ledgers specify "Ledger @@" otherwise replace the "@@" with the valid ledger code.

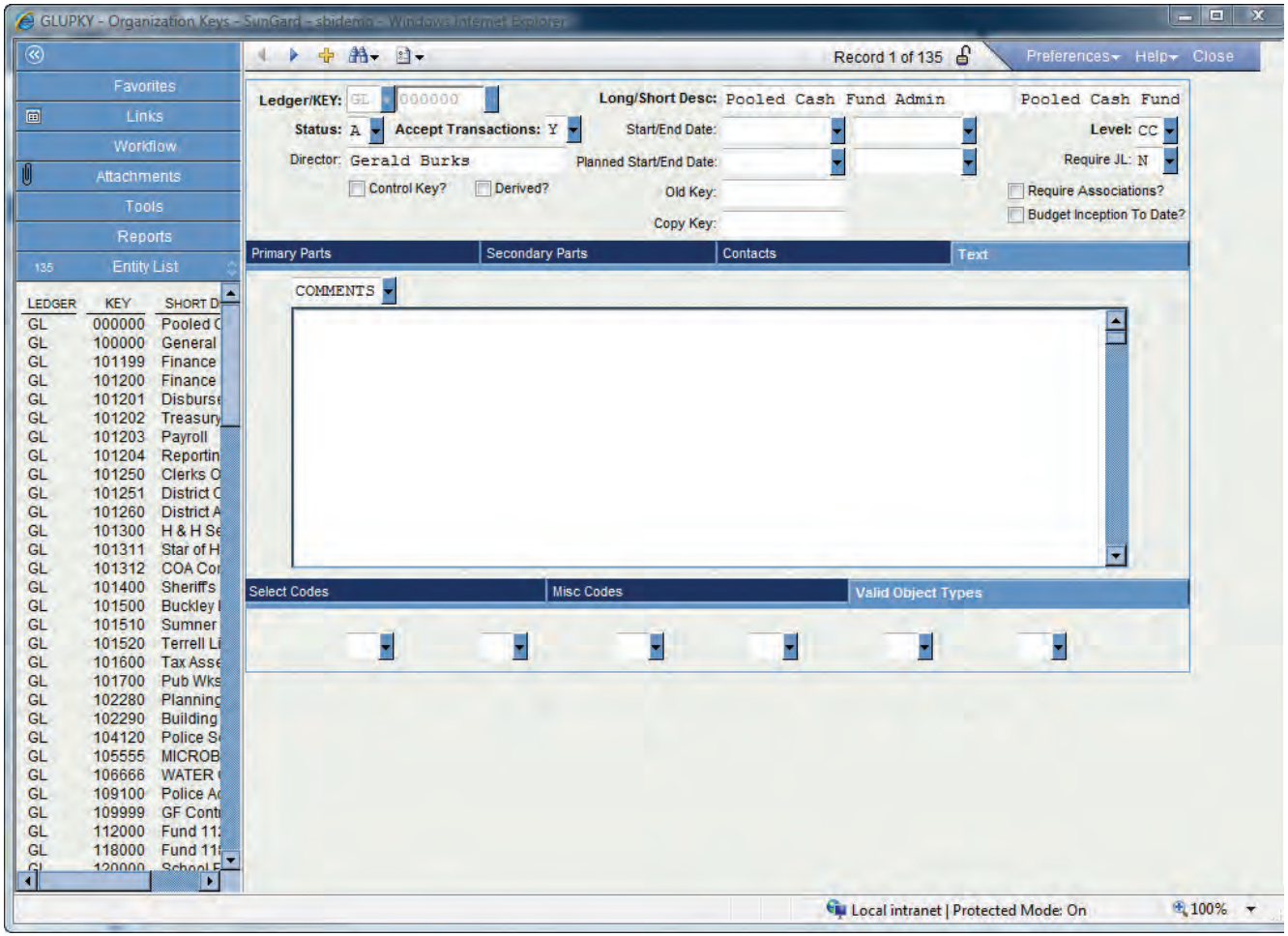
An optional date range can be specified indicating when each Key is valid. Attempts to post to a key outside its date range can be set to either warn the user or to block posting. The mask NUUPER can be used to access error messages GL479, GL473, GL480, GL497, GL481, GL482, GL483, and GL484 to control the severity of warnings/errors as well as specify grace periods. Common code GLEN-GL820 can be set to automatically begin generating warnings when Keys are used within a specified number of days prior to their end date.


GLUPKY - Organization Keys - SunGard - sbidemo - Windows Internet Explorer
Favorites
Links
Workflow
Attachments
Tools
Reports
135 Entity List
LEDGER KEY SHORT D
GL 000000 Pooled C
GL 100000 General
GL 101199 Finance
GL 101200 Finance
GL 101201 Disburse
GL 101202 Treasury
GL 101203 Payroll
GL 101204 Reportin
GL 101250 Clerks O
GL 101251 District C
GL 101260 District A
GL 101300 H & H Se
GL 101311 Star of H
GL 101312 COA Cor
GL 101400 Sheriff's
GL 101500 Buckley I
GL 101510 Sumner
GL 101520 Terrell Li
GL 101600 Tax Asse
GL 101700 Pub Wks
GL 102280 Planning
GL 102290 Building
GL 104120 Police S
GL 105555 MICROB
GL 106666 WATER
GL 109100 Police A
GL 109999 GF Contr
GL 112000 Fund 11
GL 118000 Fund 11
Record 1 of 135
Preferences Help Close
Ledger/KEY: GL 000000
Long/Short Desc: Pooled Cash Fund Admin
Pooled Cash Fund
Status: A
Accept Transactions: Y
Start/End Date:
Level: CC
Director: Gerald Burks
Planned Start/End Date:
Require JL: N
Control Key?
Derived?
Old Key:
Require Associations?
Copy Key:
Budget Inception To Date?
Primary Parts
Secondary Parts
Contacts
Text
Part Codes
Parent Control Key GL
Fund: 00 Pooled Cash
Function: 01 Administration
Override Budget Version:
Division: 000 N/A
Budget Close Code:
Depart: 0000 N/A
Over Budget Flags
Program: 0000 N/A
Action: W Level: OB
Over Allocation Flags
Action: Level: G8
Report Format: 000000
Select Codes
Misc Codes
Valid Object Types
Org Fnd Type Fnd Grp Prim/CU Activity
HARRIS GF 0001 1 1
Local intranet | Protected Mode: On
100%


GLUPKY - Organization Keys - SunGard - sbidemo - Windows Internet Explorer
Favorites
Links
Workflow
Attachments
Tools
Reports
135 Entity List
LEDGER KEY SHORT D
GL 000000 Pooled C
GL 100000 General
GL 101199 Finance
GL 101200 Finance
GL 101201 Disburse
GL 101202 Treasury
GL 101203 Payroll
GL 101204 Reportin
GL 101250 Clerks O
GL 101251 District C
GL 101260 District A
GL 101300 H & H Se
GL 101311 Star of H
GL 101312 COA Cor
GL 101400 Sheriff's
GL 101500 Buckley
GL 101510 Sumner
GL 101520 Terrell Li
GL 101600 Tax Asse
GL 101700 Pub Wks
GL 102280 Planning
GL 102290 Building
GL 104120 Police S
GL 105555 MICROB
GL 106666 WATER
GL 109100 Police A
GL 109999 GF Contr
GL 112000 Fund 11
GL 118000 Fund 11
Record 1 of 135
Preferences Help Close
Ledger/KEY: GL 000000
Long/Short Desc: Pooled Cash Fund Admin
Pooled Cash Fund
Status: A
Accept Transactions: Y
Start/End Date:
Level: CC
Director: Gerald Burks
Planned Start/End Date:
Require JL: N
Control Key?
Derived?
Old Key:
Require Associations?
Copy Key:
Budget Inception To Date?
Primary Parts
Secondary Parts
Contacts
Text
Parts 1 - 16
Parts 17 - 32
St Fund:
St Fnct:
St Dept:
Select Codes
Misc Codes
Valid Object Types
Org
Fnd Type
Fnd Grp
Prim/CU
Activity
HARRIS
GF
0001
1
1
Local intranet | Protected Mode: On
100%

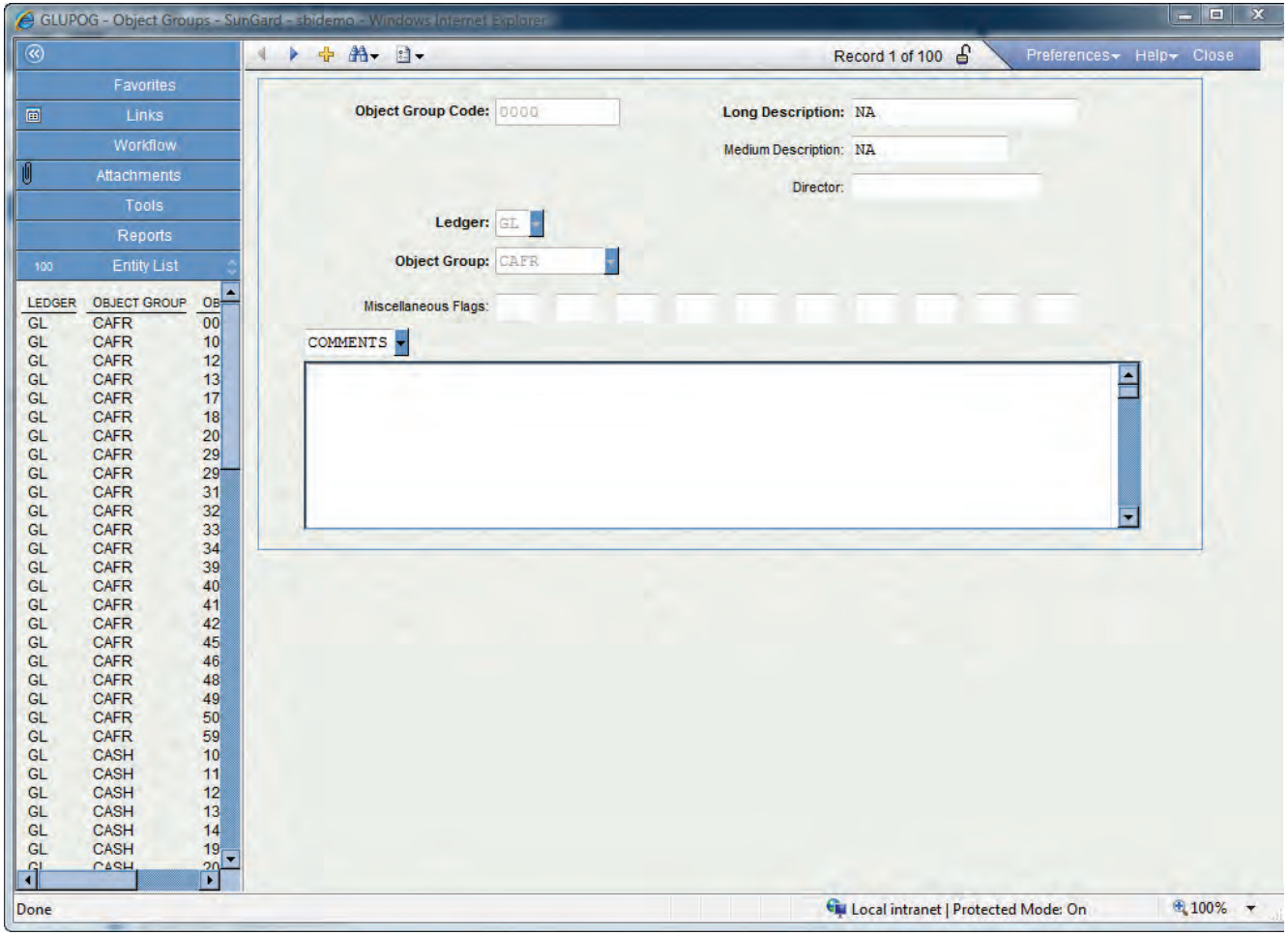

GLUPKY - Organization Keys - SunGard - sbidemo - Windows Internet Explorer
Favorites
Links
Workflow
Attachments
Tools
Reports
135 Entity List
LEDGER KEY SHORT D
GL 000000 Pooled C
GL 100000 General
GL 101199 Finance
GL 101200 Finance
GL 101201 Disburse
GL 101202 Treasury
GL 101203 Payroll
GL 101204 Reportin
GL 101250 Clerks O
GL 101251 District C
GL 101260 District A
GL 101300 H & H Se
GL 101311 Star of H
GL 101312 COA Cor
GL 101400 Sheriff's
GL 101500 Buckley
GL 101510 Sumner
GL 101520 Terrell Li
GL 101600 Tax Asse
GL 101700 Pub Wks
GL 102280 Planning
GL 102290 Building
GL 104120 Police S
GL 105555 MICROB
GL 106666 WATER
GL 109100 Police A
GL 109999 GF Contr
GL 112000 Fund 11
GL 118000 Fund 11
Record 1 of 135
Preferences Help Close
Ledger/KEY: GL 000000
Long/Short Desc: Pooled Cash Fund Admin
Pooled Cash Fund
Status: A
Accept Transactions: Y
Start/End Date:
Level: CC
Director: Gerald Burks
Planned Start/End Date:
Require JL: N
Control Key?
Derived?
Old Key:
Require Associations?
Copy Key:
Budget Inception To Date?
Primary Parts
Secondary Parts
Contacts
Text
PEID:
Address Code:
Contact Type:
Add Mode
Select Codes
Misc Codes
Valid Object Types
Org
Fnd Type
Fnd Grp
Prim/CU
Activity
HARRIS
GF
0001
1
1
Done
Local intranet | Protected Mode: On
100%


GLUPKY - Organization Keys - SunGard - sbidemo - Windows Internet Explorer
Favorites
Links
Workflow
Attachments
Tools
Reports
135 Entity List
LEDGER KEY SHORT D
GL 000000 Pooled C
GL 100000 General
GL 101199 Finance
GL 101200 Finance
GL 101201 Disburse
GL 101202 Treasury
GL 101203 Payroll
GL 101204 Reportin
GL 101250 Clerks O
GL 101251 District C
GL 101260 District A
GL 101300 H & H Se
GL 101311 Star of H
GL 101312 COA Cor
GL 101400 Sheriff's
GL 101500 Buckley I
GL 101510 Sumner
GL 101520 Terrell Li
GL 101600 Tax Asse
GL 101700 Pub Wks
GL 102280 Planning
GL 102290 Building
GL 104120 Police S
GL 105555 MICROB
GL 106666 WATER
GL 109100 Police A
GL 109999 GF Contr
GL 112000 Fund 11
GL 118000 Fund 11
Record 1 of 135
Preferences Help Close
Ledger/KEY: GL 000000
Long/Short Desc: Pooled Cash Fund Admin
Pooled Cash Fund
Status: A
Accept Transactions: Y
Start/End Date:
Level: CC
Director: Gerald Burks
Planned Start/End Date:
Require JL: N
Control Key?
Derived?
Old Key:
Require Associations?
Copy Key:
Budget Inception To Date?
Primary Parts
Secondary Parts
Contacts
Text
COMMENTS
Select Codes
Misc Codes
Valid Object Types
Misc Code 1
Misc Code 2
Misc Code 3
Misc Code 4
Misc Code 5
Misc Code 6
Misc Code 7
Misc Code 8
http://bsnug79/ifas7/OptionsBar/OptionsBar.html#
Local intranet | Protected Mode: On
100%


GLUPKY - Organization Keys - SunGard - sbidemo - Windows Internet Explorer
Favorites
Links
Workflow
Attachments
Tools
Reports
135 Entity List
LEDGER KEY SHORT D
GL 000000 Pooled C
GL 100000 General
GL 101199 Finance
GL 101200 Finance
GL 101201 Disburse
GL 101202 Treasury
GL 101203 Payroll
GL 101204 Reportin
GL 101250 Clerks O
GL 101251 District C
GL 101260 District A
GL 101300 H & H Se
GL 101311 Star of H
GL 101312 COA Cor
GL 101400 Sheriff's
GL 101500 Buckley I
GL 101510 Sumner
GL 101520 Terrell Li
GL 101600 Tax Asse
GL 101700 Pub Wks
GL 102280 Planning
GL 102290 Building
GL 104120 Police S
GL 105555 MICROB
GL 106666 WATER
GL 109100 Police Ac
GL 109999 GF Contr
GL 112000 Fund 11
GL 118000 Fund 11
Record 1 of 135
Preferences Help Close
Ledger/KEY: GL 000000
Long/Short Desc: Pooled Cash Fund Admin
Pooled Cash Fund
Status: A
Accept Transactions: Y
Start/End Date:
Level: CC
Director: Gerald Burks
Planned Start/End Date:
Require JL: N
Control Key?
Derived?
Old Key:
Require Associations?
Copy Key:
Budget Inception To Date?
Primary Parts
Secondary Parts
Contacts
Text
COMMENTS
Select Codes
Misc Codes
Valid Object Types
Local intranet | Protected Mode: On
100%

2.2.3 Object Group Definition

After entering Org. Key Definitions, go to the GLUPOG form. This is the form on which valid code values for each of the Object Groups are defined for a given Ledger. In much the same manner as the user-defined Organization Key Parts Codes, the Object Group Codes will be defined here. All Object Group Codes must be defined before defining Object Codes. Each value entered has a long and short description and is associated with a previously defined Object Group. A Ledger Code must be indicated since the system will automatically check to see that the Object Group has been defined.



2.2.4 Object Code Definition

After all Object Group Codes have been defined, go to the GLUPOB form. This is the form on which each Object Code is defined and associated with the various background codes established for the Object Groups.

Type codes are defined in the system to provide broad groupings of different Object Code classifications. Appropriate codes are: 'AS' - Assets, 'LI' - Liabilities, 'FB' - Fund Balance, 'RV' - Revenue, 'XP' - Expenditures, 'TI' - Transfers In and 'TO' - Transfers Out. They are used for report extractions, transaction entry control, and the fiscal year-end process. With the **Require Assoc?** field, each Object Code may be required to be associated with an Org. Key before transactions may be entered against it. This insures that the resulting Chart of Accounts will be free of undesirable accounts.

Individual Object Codes must have a Balance Type, 'DR' Debit or 'CR' Credit. When defining contra objects, maintain the same Balance Type as the objects that they are contra to. For instance, a contra revenue account should have a Balance Type of 'CR'. This allows the actual balances in contra accounts to be reported with a minus (-) sign, making them easily recognizable.

While most Object Codes will not have a Start/End Date, this is helpful when changing accounting practices and the user does not wish transactions to be keyed against an unused or invalid Object Code value. An optional date range can be specified indicating when each Object Code is valid. Attempts to post to an Object Code outside its date range can be set to either warn the user or to block posting. The mask NUUPER can be used to access error messages GL479, GL473, GL480, GL497, GL481, GL482, GL483, and GL484 to control the severity of warnings/errors as well as grace periods. Common code GLEN-GL820 can be set to begin automatically generating warnings when Object Codes are used within a specified number of days prior to the end date.

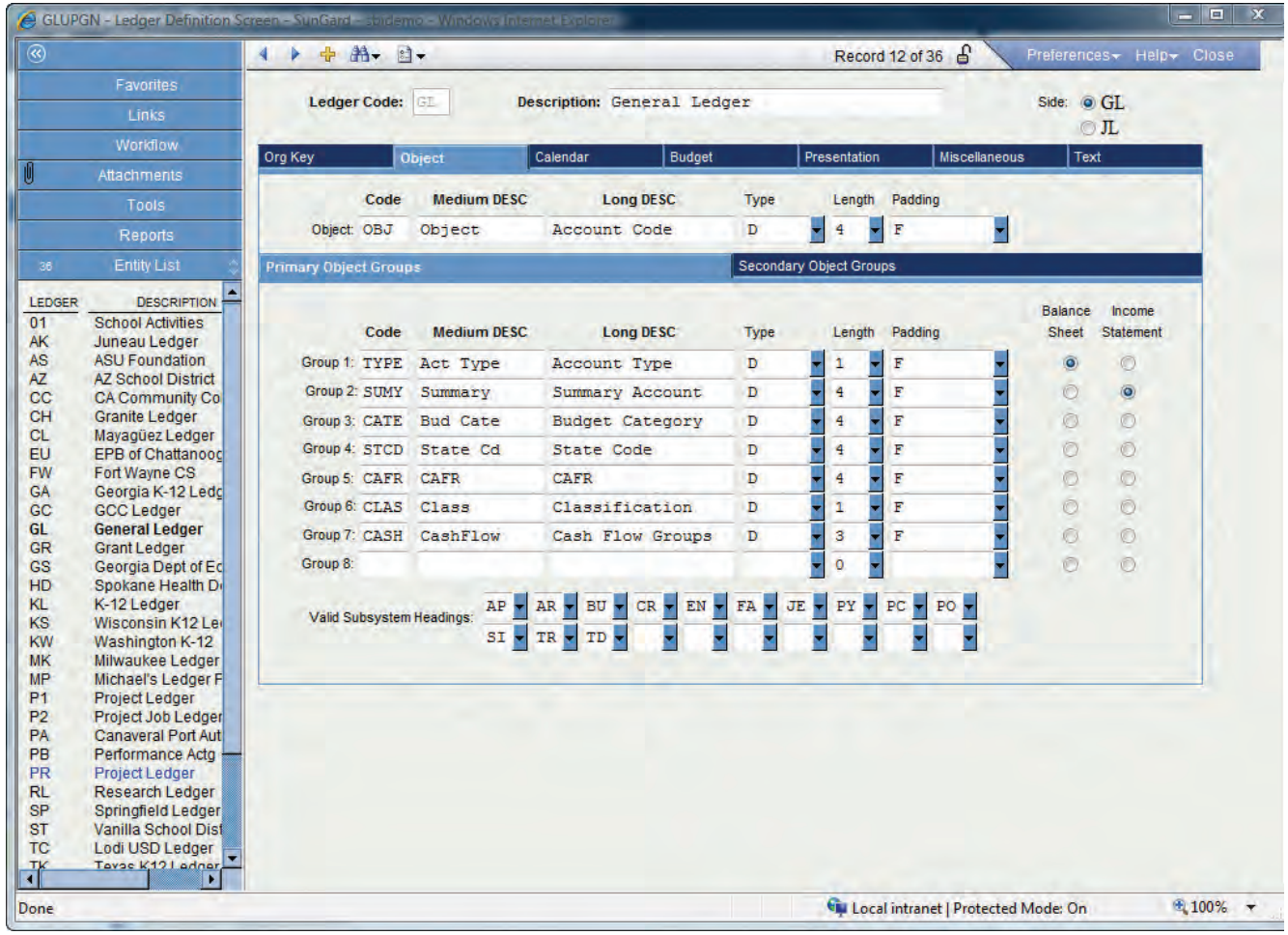
Under certain circumstances, it may be desirable to require a JL Account when a specific Object Code is entered at data entry time. This can be accomplished by placing a 'Y' in the **Require JL?** field.

There may be times when an individual is assigned responsibility for one or more object codes such as an accountant who prepares inventory or investment schedules. In these cases, they may be assigned as a Director and easily extract information for analysis purposes.

When the Object Groups were defined in **GLUPGN**, the Object Code Subsystem Field Titles were defined. They now appear above each of the boxes beside the **Allow Subsystem Transaction** area. The system assumes that each Object Code will accept transactions from all subsystems. However, if this is not desired, place an 'N' under each subsystem from which transactions should NOT be accepted.

Each of the defined Object Groups is echoed on this form and must have an appropriate Object Group Code entered. Upon pressing the 'ENTER' key, the Obj. Group code descriptions will be displayed for visual verification.

As in the definition of Org. Keys, the Old Code from the previous system may be entered for historical audit purposes and for mapping transaction rollover to IFAS.

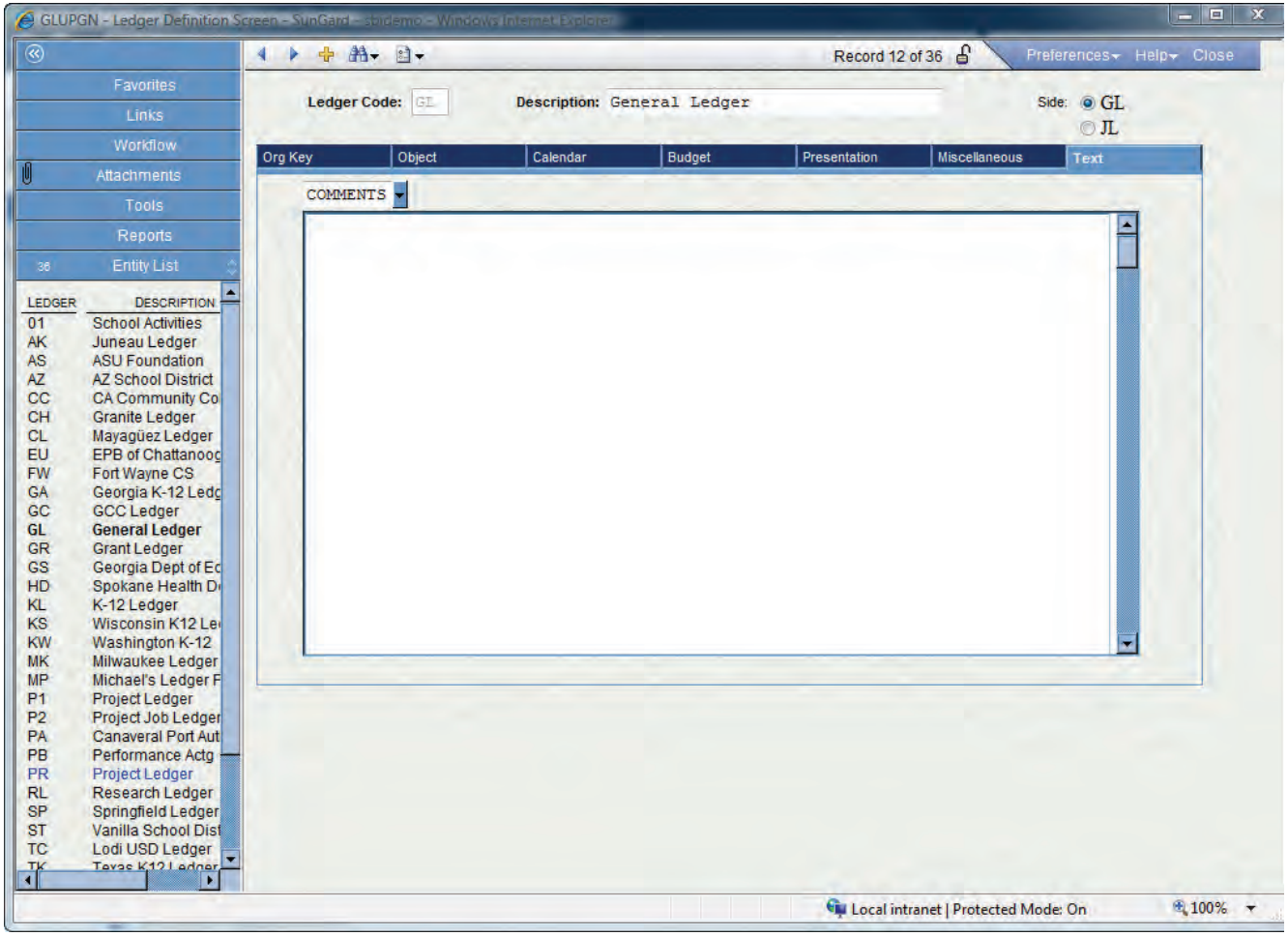


2.2.5 Associated Text for Organization Keys, Object Codes, Key Parts, and Object Groups

This form allows free form text to be entered which relates to individual Org Keys, Object Codes, Key Parts, and Object Groups. The text entered on this screen may be optionally printed on the GLREFLKY, GLREFLOB, GLREFLPT, and GLREFLOG file listing reports.

At the top of this screen, enter the type of item you wish to enter text for; KY for Org Key, OB for Object Code, PT for Key Part, or OG for Object Group. After the type is entered, a valid value for that type must be entered.

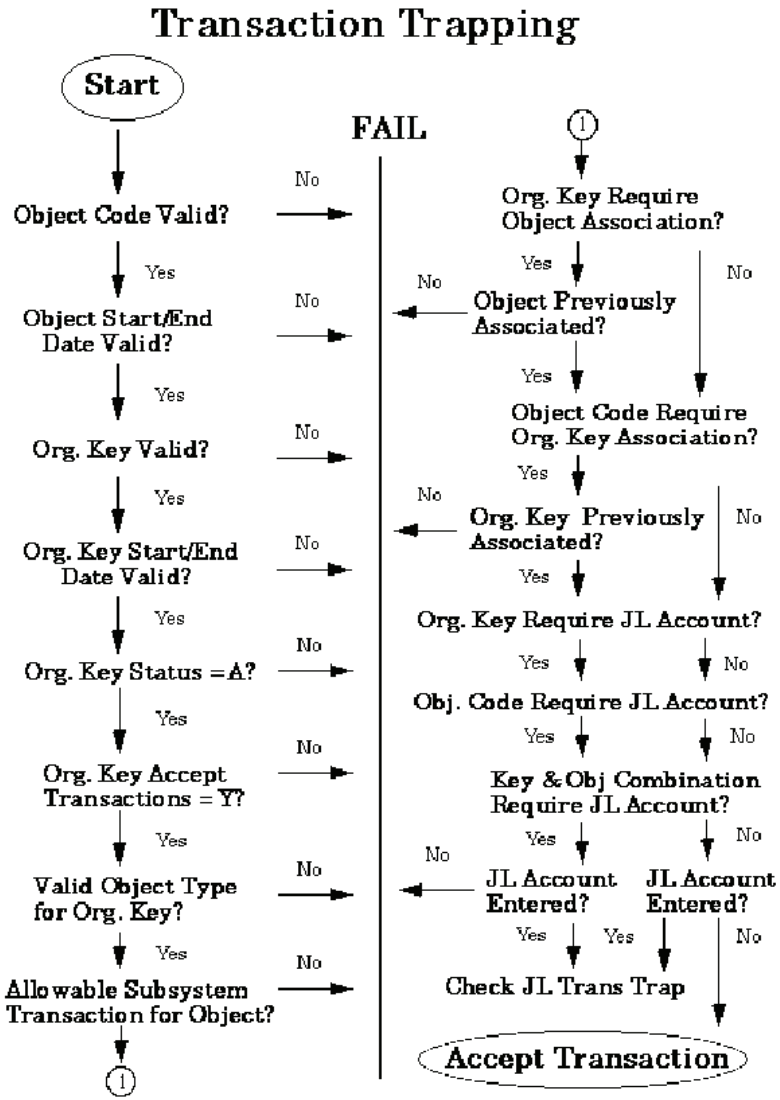
In the Line column, put a line number. In the text column on the same line, enter text describing the value entered at the top of the screen. An unlimited number of lines may be entered in this manner. A line number must be entered for text to be saved.



2.2.6 Transaction Trapping

The term 'Transaction Trapping' refers to the process used at data entry time for IFAS to automatically stop any transactions which should not be entered against a specific account. Many of the flags and codes entered when defining Org. Keys and Object Codes are used by the system to perform the trapping based on user defined requirements. The following chart identifies the fields on the definition forms and the sequence in which they are checked to perform the 'trapping'. At any specific point if a 'trap' fails, the user is given an appropriate error message and

additional data entry for that item is not allowed until the problem is corrected. The 'trapping' will also occur on both the 'GL SIDE' and 'JL SIDE' if so defined by the user. In that instance, the "GL SIDE" is checked first and then the 'JL SIDE'.



2.2.7 Budget Checking

The term ‘Budget Checking’ refers to the process used to compare actual transaction activity against a predefined named version of the budget. Based on the type of 'checking' desired, the data entry person may receive a warning, an “authorization required” screen, or *a complete* block if the transaction entry would cause a budget overrun to occur. From all IFAS subsystems, budget checking will include the following types of data and compare it to the appropriate budget: Purchase Requisitions, Unencumbered Purchase Orders, Encumbrances, Open Accounts Payable, and existing GL and/or JL account balances. The following chart outlines the sequence in which budgets and/or portions of the GLDB structure are reviewed to determine the level of budget checking which is required. While at first this may seem overly complex, there are only four basic steps:

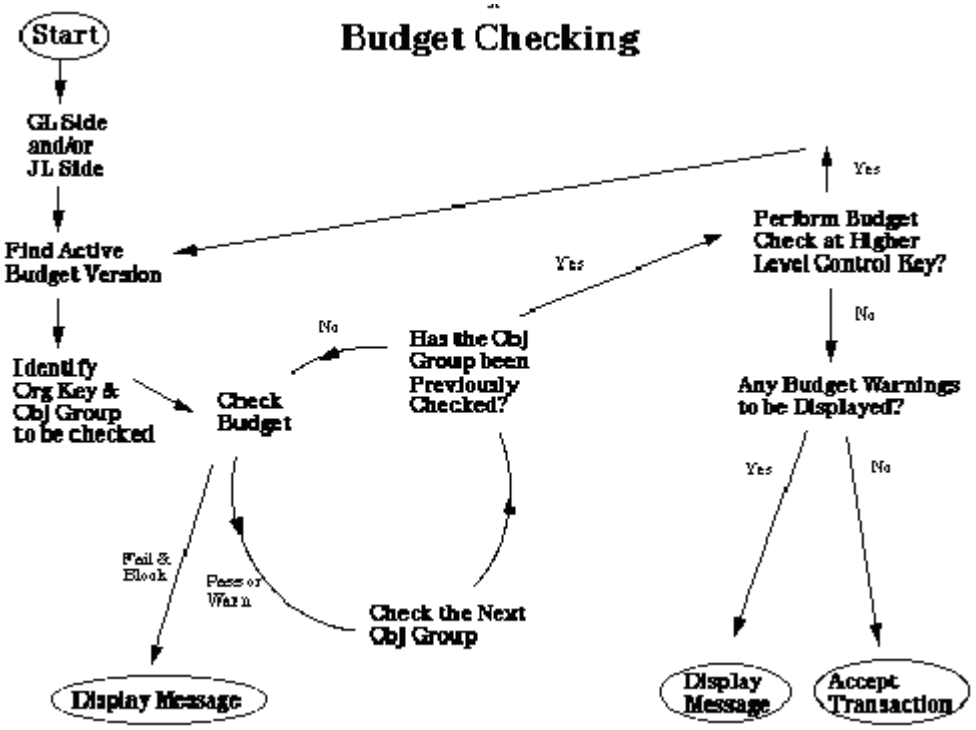
1. Has ‘Budget Checking’ been defined for this account?

2. Which named version of the budget is to be used for the comparison to actual?

3. What Object Group checking is desired and is there more than one Group to be checked?

4. Is there a 'Control Key’ defined at a higher level budget within the organization which must be checked?

Refer to, 'Budgeting' for detailed strategies for Budget Control and Checking.



2.3 Optional System Features

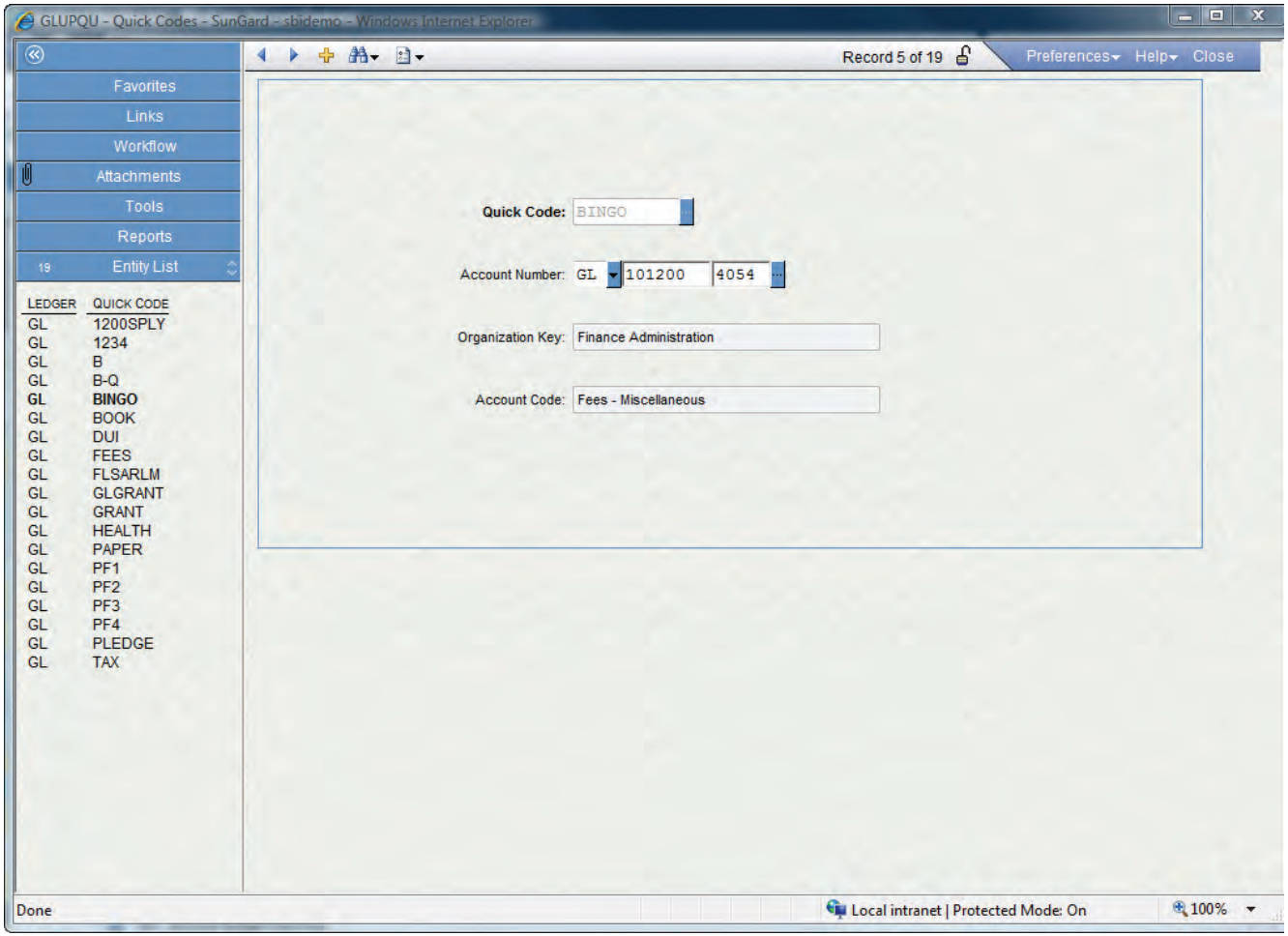
In prior sections, this user guide has discussed use of the Org. Key and Fully Qualified account methods. A final, short procedure is the use of Quick Account Numbers. In this instance, the single combination of an Org. Key and Object Code (account) is associated with a unique number. Another feature is the ability to establish rules for the creation of Org. Keys which insures that only appropriate combinations of the Org. Parts may be made.

2.3.1 Quick Account Numbers

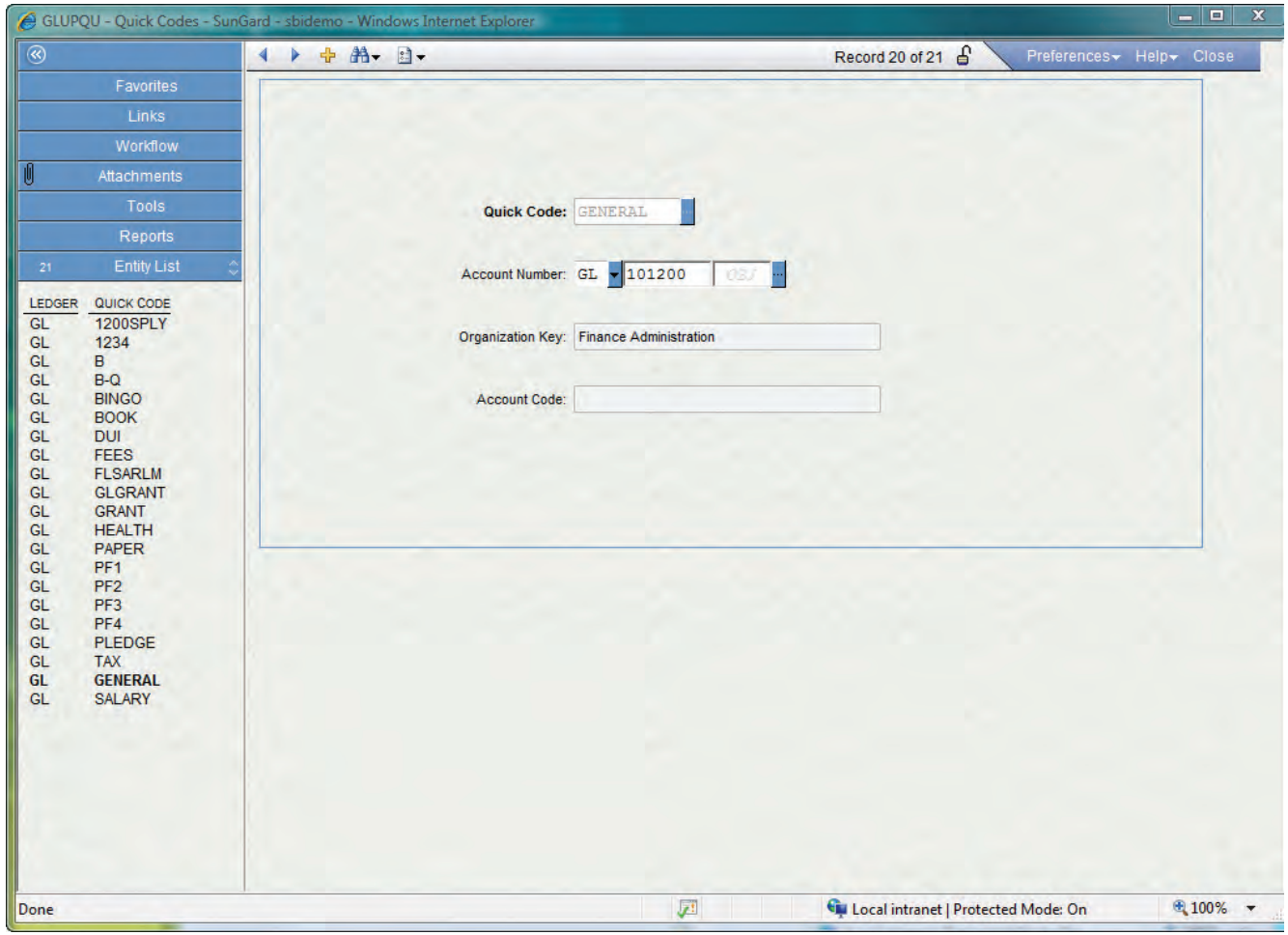
Under some circumstances, it may be easier to enter a single number for an account rather than requiring entry of the Org. Key-Object or the Fully Qualified Account. This might be in a department which has very few accounts and would like to enter only a three or four-digit number. Another example might be in accounts where there is a high level of data entry activity such as Supply or Telephone accounts. In these instances, a shortened account number, known as the Quick Account Number, may be created. This does not change the standard input format

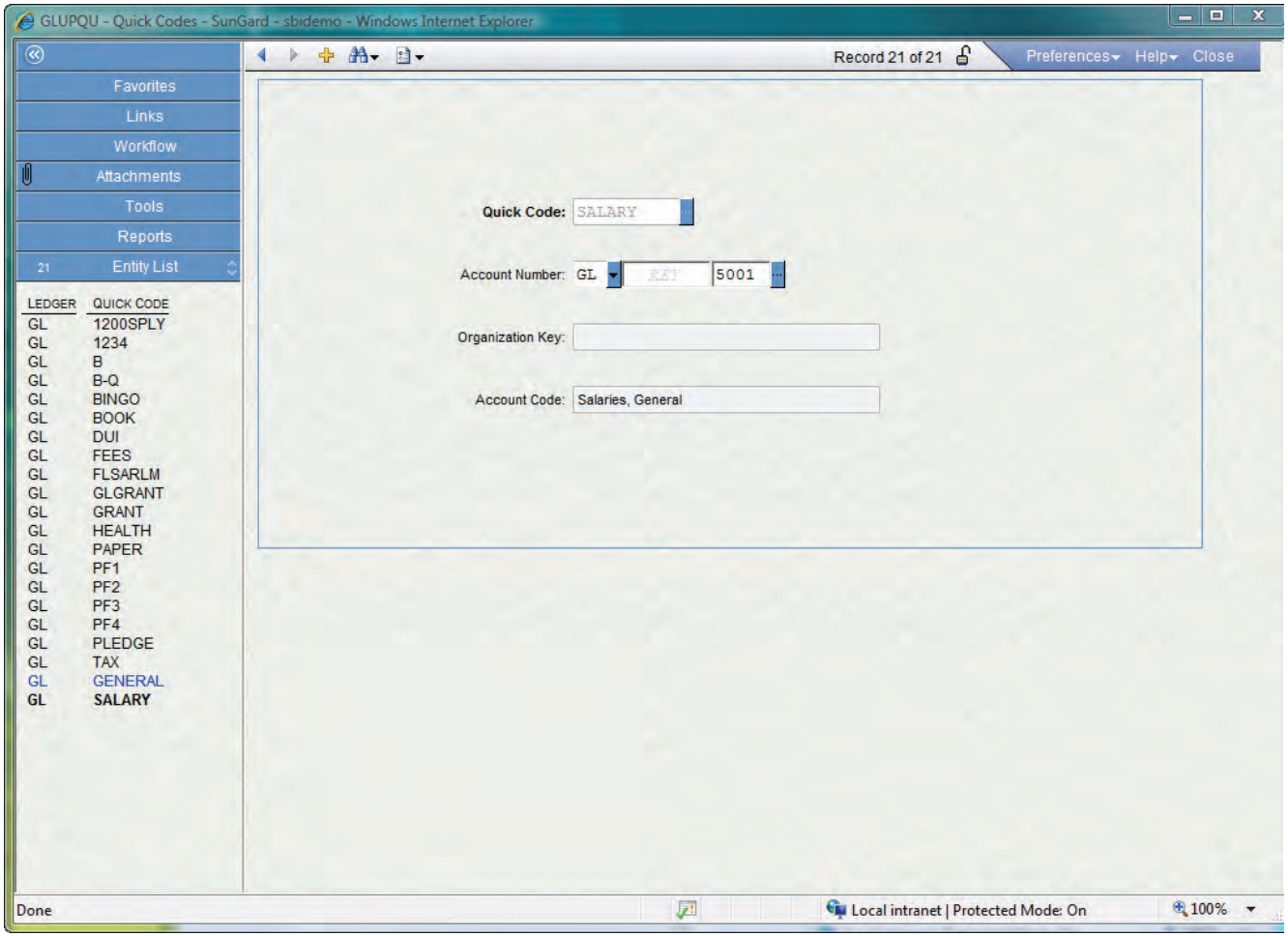
but provides an optional override method for account entry. In the example below the GL Ledger, Quick Number '305' is used in the place of Account '1012005001'.

To define a Quick Account Number, from the main menu key **GLUPQU**. The following form will be displayed.



Optionally a string may be defined in the quick key account number field. This string must be prefixed by a *. All occurrences of the quick key will be replaced with the string value at date entry time. For example: You define quick key 'GENERAL' which maps to string '*101200' and quick key 'TRAVEL' which maps to string'*5001'. When 'GENERAL-TRAVEL' is entered in an account number field on a data entry screen, it will be translated to 1012005001.





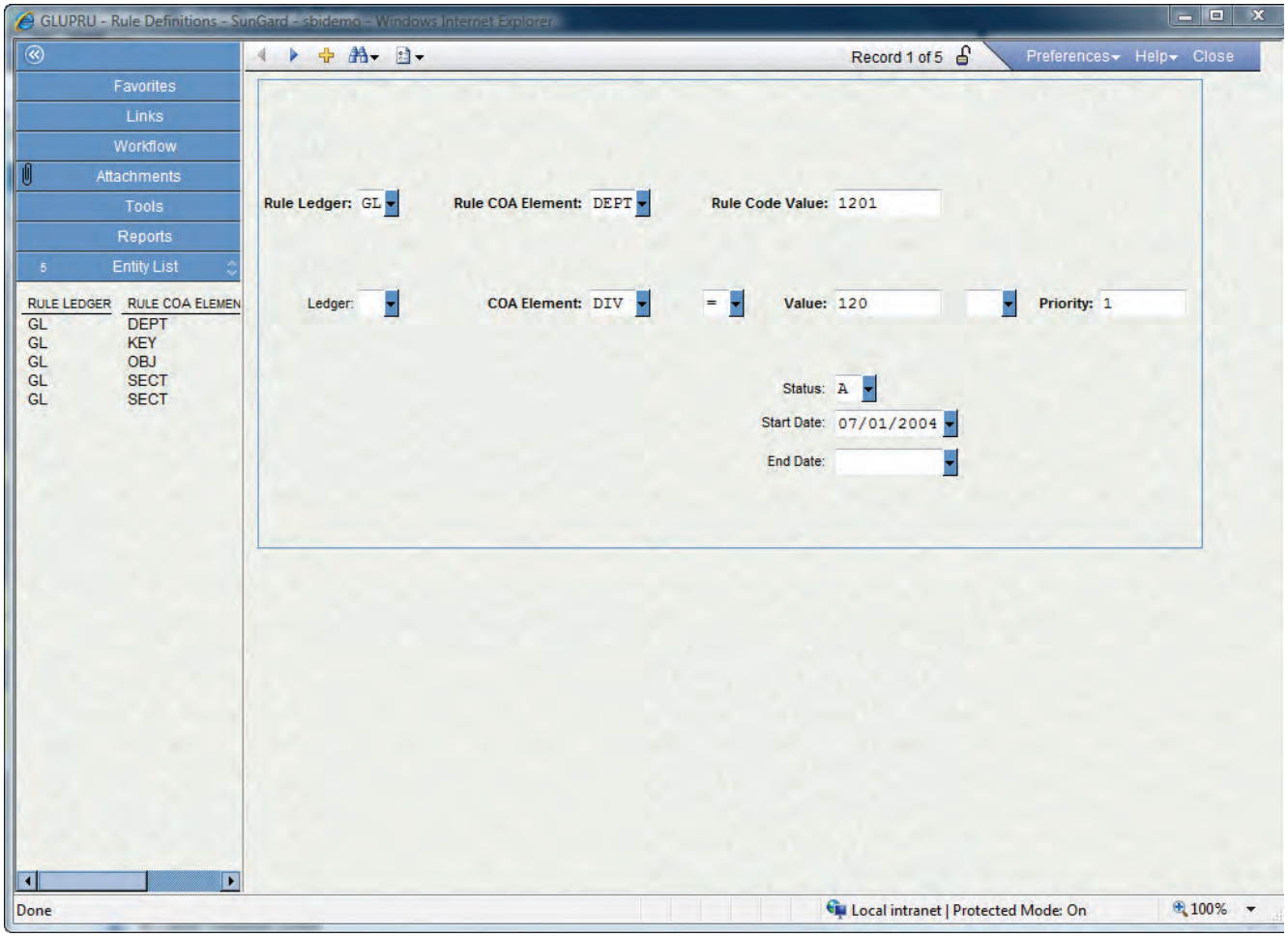
2.3.2 Rule Logic for Key Creation

Many times it is important to establish rules within the software that insure the correct codes are used when creating Org. Keys, Object Codes, or Accounts. For example, whenever an Org. Key is created within the ADMN Function it must be associated with the FUND '10' and '30', and OFCR 'VPFA'. This may apply to either its definition within the ORG. KEY DEFINITION FORM, GLUPKY, or when using the Fully Qualified Account method, at data entry time with the automatic Org. Key generator. By defining these types of rules or logic, IFAS will require that the correct combination of Org. Part Code Values will be entered. This increases the level of internal documentation for the creation of account information and helps to insure the validity of the account reporting structure.

In creating rules related to a specific item, the user should think of the following syntax:

For the given Ledger, Org. Key, Org. Part, Object Code, Object Group, Org. Key Select Code, Org. Key Misc Code, or Object Type, the following Code(s) may be associated together and will be reviewed in the order of priority defined.

To define the rule logic, from the main menu key GLUPRU. The following form will be displayed:



2.4 JE Approval Definitions

The Journal Entry (JE) approval function in the General Ledger (GL) subsystem functions in a manner similar to the approval functions in the Purchasing (PO) and Accounts Payable (AP) subsystems. These similarities include distribution of regular set files from the DATA sub-directory into database tables referred to as the "approval area". The file gets automatically moved from DATA to POSTED upon distribution

into the “approval area”. Approvals are done only while records are in the “approval area”. Configurable user approval capabilities and classes determine the approval hierarchy and “routing”. Fully approved “references” (JE-IDs) are allowed to be distributed into the GL database.

Other approval masks allow distribution of JE sets into the approval system (GLJEAPDA), run a JE set proof on a JE-ID in approve (standard - GLJEAPBP, alternate security - GLJEAPBX, and one-sided - GLJEAPOB), approve (reject or update) JE-IDs in the approval system (standard -GLJEAPUB, one-sided - GLJEAPOJ, quick - GLJEAPQU, wide account - GLJEAPUW, and alternate security - GLJEAPUX), report approved JE-IDs in the approval system awaiting posting (GLJEAPRA), report JE-IDs in the approval system needing approvals (GLJEAPRN), and distribute approved JE-IDs to posted (standard - GLJEAPDS, multi-date GLJEAPDDM, one-sided multi-date GLJEAPOM, and one-sided GLJEAPOS). An approvals archiving and purging utility (GLUTSUAA) is also provided to allow accumulated approval history to be manually purged based on completion date and selection criteria. Completed approval history is saved upon distribution into GL if the short description of the GLAP-DEFAULT common code is set to “SAVE”.

Two approval class groups shall be required on each JE transaction. The first applies to all transactions collectively within the JE-ID while the second is specific to a particular JE transaction. The first approval class group shall be the same on all transactions within the JE-ID. The approval class groups shall identify which approval code classes are applicable to each transaction and will establish their hierarchical order. The approval code classes, in turn, shall identify which approval codes should be checked for each transaction and establishes hierarchical order. The use of two different approval class groups allows segregation of single transaction tests from multiple-transaction tests. The two-level hierarchical grouping (approval class groups and approval code classes) assists in organizing and simplifying maintenance of the approval configurations as well as establishing a hierarchy.

Approval codes are specific tests applicable to either individual transactions, or all the transactions collectively within a JE-ID. Their definition contains a set of conditions that may either be true or false. If the conditions do not apply (e.g., the test result is false) the approval code is satisfied and no special approval of it is necessary. If, however, the specified conditions apply (e.g., the test result is true), a special condition exists which requires approval by the appropriate individual(s).

User capabilities shall be defined for each user identifying their capabilities in three ways: the approval class groups they are allowed to use, the approval codes they are allowed to approve, and any special capabilities they are allowed to have. Approval class groups for each user determine which JE transactions users are allowed to access. Each one is accompanied by an “M” for modify or “A” for approve only. This controls what the user can do to JE sets in the Set Editor. Two additional fields specify the default JE-ID level and transaction level approval class codes for each particular user. Approval codes for each user constitute the list of test conditions they are allowed to approve. Each one is accompanied by a “P” if the user is a primary approver or an “S if the user is a secondary approver.

Additionally for each user, twenty special capability flags are included for future expansion. They could be used to determine which functions specific users will be allowed to perform. At the current time, only two special capability controls are implemented; the rest are included and reserved for future use. The first user special capability flag is labeled “RejDsk” and is displayed as a 1 character wide field. Only the values “Y” and “N” are valid. If the “RejDsk” flag is set to “Y”, rejections in approval for files created by such a user shall be rejected completely out of the approval area and back into a JE set file in the DATA directory on the hard disk where the user can edit it as they did originally. If the flag is “N”, rejections in approval will clear the JE-ID or transaction level approvals, but leave the transactions in the “approval area”. The second user capability flag is labeled “1stApr” and is displayed as a 1 character wide field. Only the values “Y”, “A”, and “N” are valid. If the

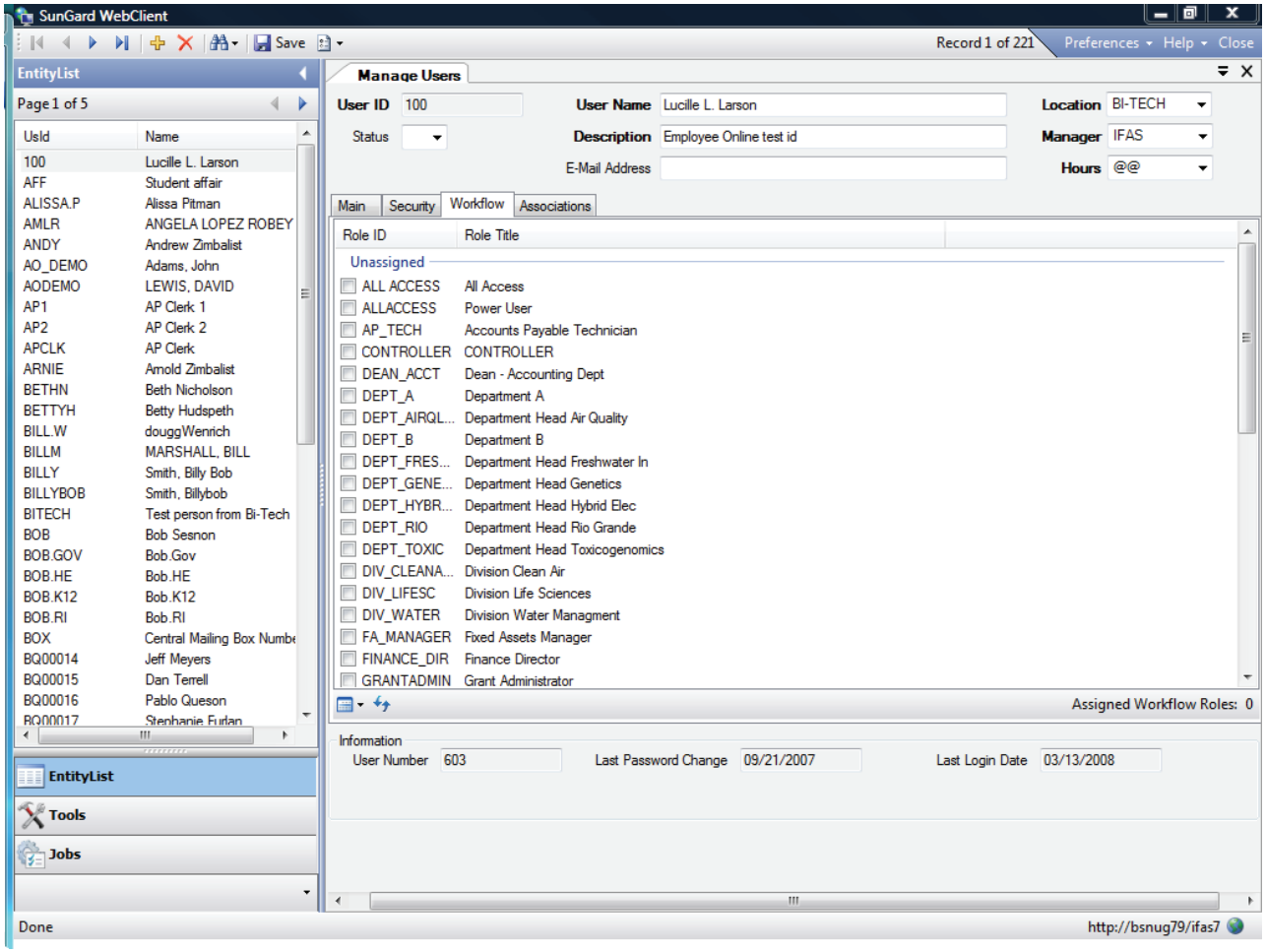
“1stApr” flag is set to “Y” or “A”, the creator automatically becomes the first approver, followed by those defined by the approval groups. Immediately after the file is distributed into approve, the first approval can automatically created and assumed by setting the flag is set to “A”. In this case, the creator would only have to approve their own work first if it has been rejected or modified, but not when it is distributed into approve the first time. If the flag is “N”, there is no special processing to make the creator the first approver, and the approvals follow those defined by the approval groups.

Approval class groups are four characters in width and may contain only the digits 0 - 9 and the uppercase letters A - Z. No other characters are allowed. When entering them on the user capabilities screen of GLUPAP, a single @ (wildcard) sign may be used to match one character position, or @@ (double wildcards) may be used at the end to match any/all occurrences. For example, “W@@” will match all approval class groups starting with W, and “SD@@” matches all occurrences starting with SD. When wildcards are used, the hierarchical order within a wildcard set is assumed to be alphabetical.

Approval Hierarchy shall be determined as follows. Each approval class group shall be expanded into approval code classes in the order defined on the approval class group. Each of the approval code classes shall be expanded into approval codes in the order defined on the corresponding approval code class. Starting from the beginning of this list of approval codes, the first occurrence of each will be retained in place and subsequent occurrences shall be culled. The resulting list is the approval hierarchy order of approval codes. There is one exception to this: If the second user capability flag (labeled “1stApr”) is set to “Y” or “A”, the creator automatically shall become the first approver, followed by the rest of the hierarchy. It the flag is set to “A”, this approval shall be automatically satisfied immediately after a file is distributed into approve, but not after rejection or modification. Thus, the creator would only have to approve their own work first after it has been rejected or modified, but not when it is distributed into approve the first time. If the flag is “N”, there shall be no special processing to make the creator the first approver, and the approvals follow those defined by the approval groups.

To setup Journal Entry Approval Definitions, from the main menu access mask GLUPAP. The first form will be displayed.

2.4.1 S GL Approval User Definitions



2.5 Default Account Mapping

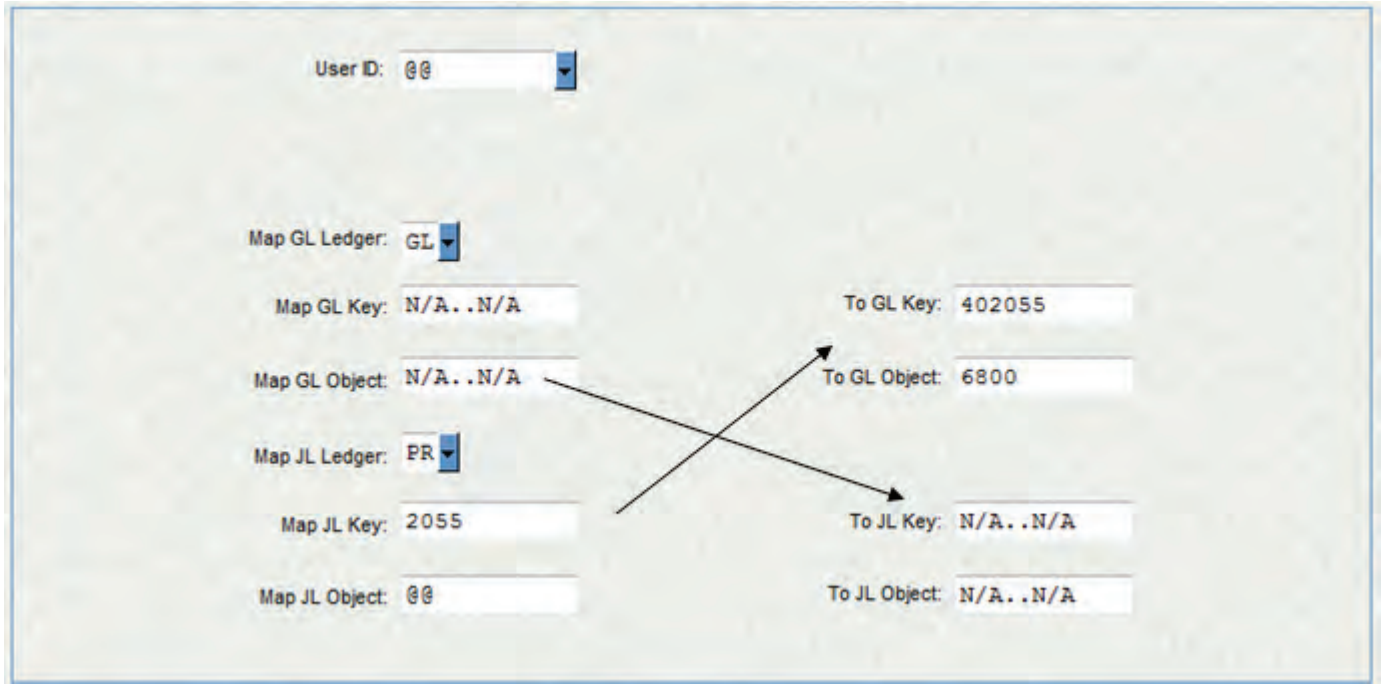
The DEFAULT ACCOUNT MAPPING screen is used to establish a link between GL Keys/Objects and JL Keys/Objects. The top portion of the screen, "When USER xxxx enters a combination of:", instructs the system to look at the user's entry in the Account field on a data entry screen to see if it matches the values entered in the "combination of" fields.

For example: We want to allow the user to simply enter the JL org key of 10102 and automatically have the GL object and the entire GL account set as defaults. The "combination of" field will contain only a JL org key of 10102. The 'default' section would be filled out with the GL key and object and JL object.

The DEFAULT ACCOUNT MAPPING screen below is shown in BROWSE-mode as an example.

Note: The system option flag 'DefAct.' on the GLUPGN screen (1) must be set to “Y” for the default account mapping to go into effect.

Optionally, the GLDF-OBJTYPE common code may be used to allow default mapping to be limited by the object type of the GL object code to be mapped. If the object type does not exist in the concatenated list of object types stored on the common code, the entire account GL/JL will not be mapped. For more information, see the description of this common code in the Common Codes user guide.



3 Journal Entries

3.1 General Ledger Journal Entries

The creation and distribution of Journal Entries into IFAS is a three step process:

1. Create or Update a Journal Entry Set File

2. Obtain a JE Set Proof File Listing

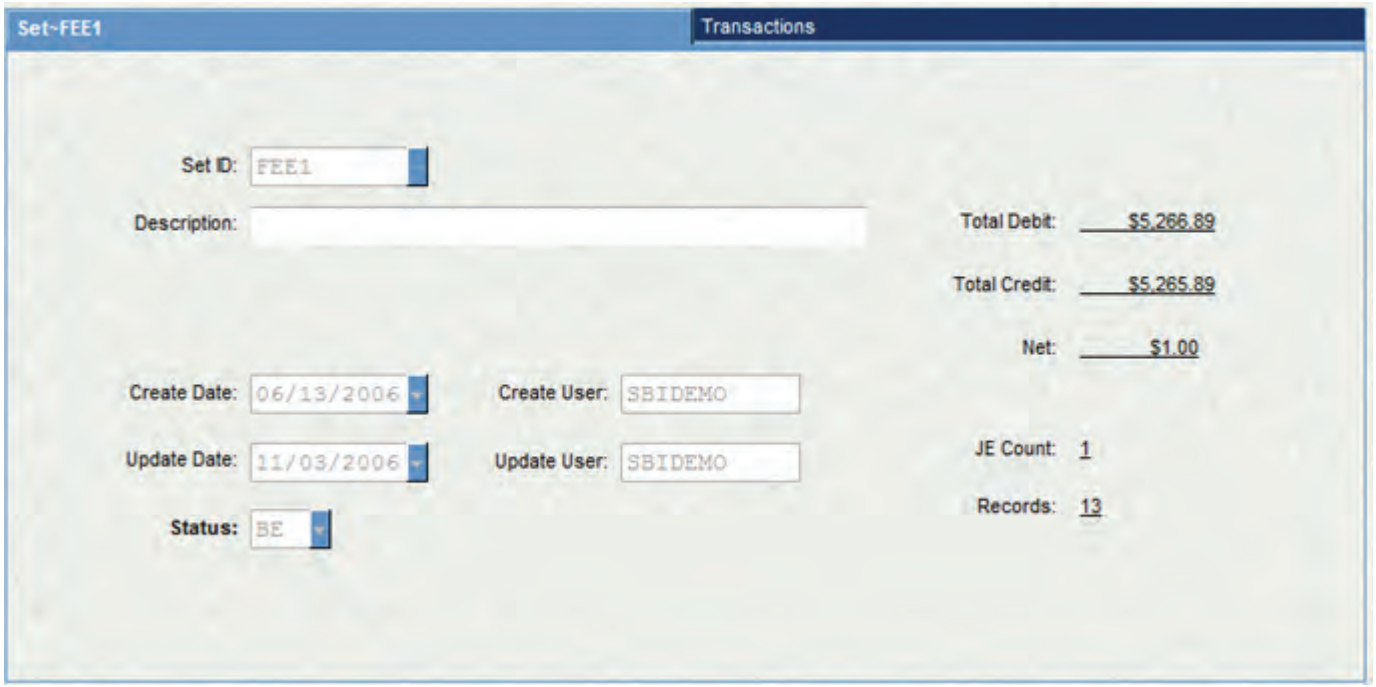
3. Distribute the JE Set File

No time limit is imposed by IFAS on when each step must occur but each step must occur in the order given. Step 2 may be performed as often as desired. Set files are created in the DATA group and remain in the DATA group until distribution. Once a set file is distributed, it is then moved to the POSTED group thus protecting it from accidental distribution a second time. A User can retrieve the Set File in the POSTED group at any time if it is necessary. This is accomplished by qualifying the Set File Name with POSTED. Access to the POSTED group is controlled through user security. General Ledger Journal Entries are prepared on the forms shown on the following pages. To create or update a General Ledger Journal Entry, key in GLJEUB, GLJEUI, GLJEUN, GLJEUW, GLJEUX, or GLJEDP. See section 1 in this chapter for more information on each of these masks. Each Set File must be given a unique name. The file name may be a maximum of eight (8) characters and must start with an alphabetic letter. For instance, the following would be an appropriate Set File Name:

JEHB0125

"JE" = Journal Entries, "HB" = persons' initials, and "0125" = 125th set since the beginning of the fiscal year. A set file may contain several sets of entries in the same file by changing the code in the Set ID field. Each Set ID identifies separate sets of entries within the set file.

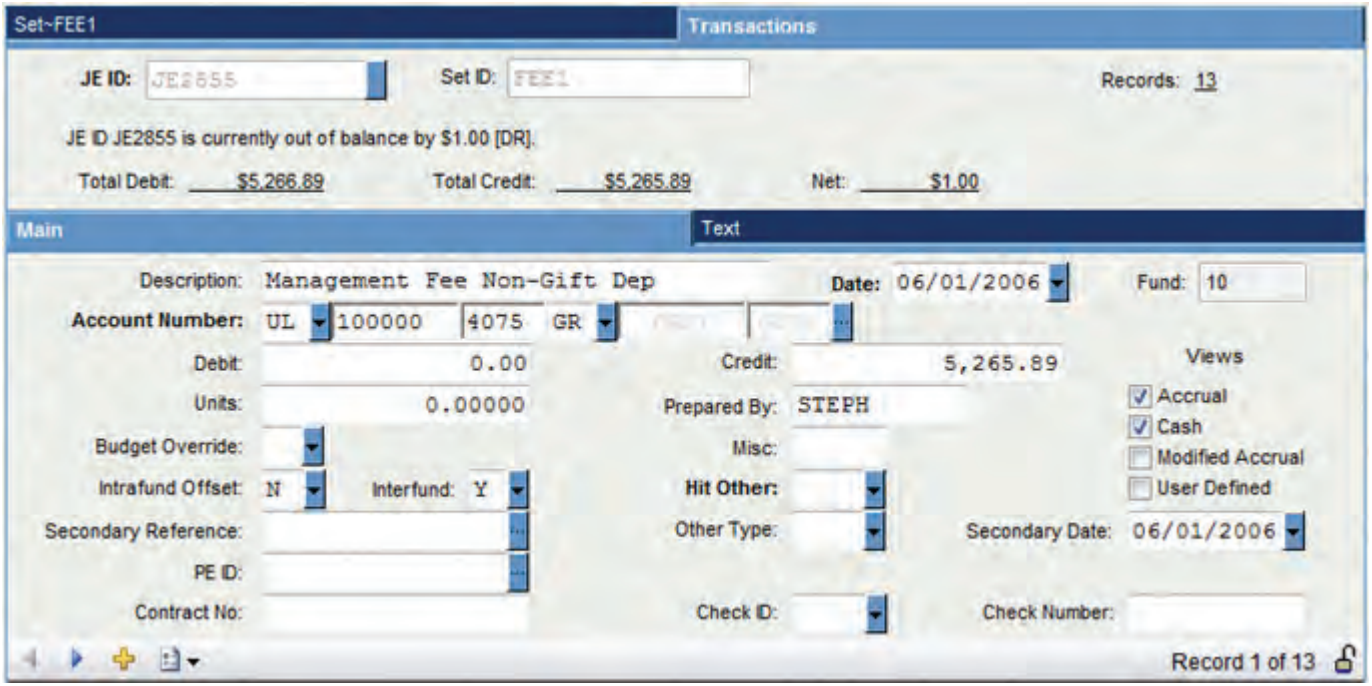
In order to create a set of transactions, GLJEUB and press return. The following screen will be displayed:



The user has two options at this point. A new JE set can be created, or an existing set may be accessed. To create a new set, simply key in the new set name and press ENTER. The system will say, “File does not exist; press ENTER to create.” If the name is keyed correctly, press ENTER once more to create the new file. The user will then be taken to the Journal Entries screen. To access an existing set, click on the find/search button and type in the set ID, or click on the set id on the left under entity list. The selected set will appear on the screen.

The entry of Account transaction information is based on the input format which was defined on the ORGANIZATION PARTS DEFINITION form. The generic input format is "glkey globj/jlkey jlobj". When entering both General Ledger side (GL) and Job Ledger side (JL) accounts for a transaction, it is necessary to place a slash (/) between the two account numbers. For instance, in the sample structure the following input formats were identified: GL Ledger = "K-O" and PR Ledger = "K-O". These would be entered in the Account field as "101200-5001/200-DESIGN".

While the normal processing of Journal Entries will be for dollar amounts, the system will also accept transactions based on units. Units might be helpful in tracking the number of labor hours expended, addition of units to endowment principal, or kilowatts charged for electricity usage. Any value entered in the "DEBIT" or "CREDIT" field preceded by a "U" will be distributed into the General Ledger as units. All units may be entered with accuracy up to five (5) places to the right of the decimal point. Units may be associated on the same transaction with dollar amounts or separately without dollar amounts. If units are entered on the credit side, they will be posted to the account as a negative value with a minus sign. An entry of this type of information would appear as follows:



Each set of balanced entries must have a unique Journal Entry ID (JEID). While one description may be given for all entries within a specific JEID, the system also allows a transaction description for each different account or any other user desired grouping of entries within a JEID. This is accomplished by entering the JEID, Date, Description, and Account information. Next, in the "Last?" field, place an "N" indicating that this is not the last item for this JEID. The system will accept the account information, clear the Account fields, and allow the user to change the JE Description field and enter additional information.

In addition to checking for equal debits and credits, the system also checks to make sure that the ""Intra-Fund Clearing" accounts are in balance and the "Inter-Fund Clearing" Claim on Cash postings are in balance. This feature is controlled by the use of the "OF" or Offset column. IFAS will not generate entries to the ""Intra" or "Inter" fund accounts if they are not needed. For a discussion of these automatic entries, please refer to Section 7 of this User Guide. The default Offset values can be easily changed to match specific clients desired implementation of the automatic, self-balancing processes.

When a set is created using the Intra-Fund Journal Entry Set Editor all funds within the set must be balance and the Offset code must always be "NN". No manual inter-fund offsetting or automatic offsetting of any kind will be allowed.

When a set is created using the Inter-Fund Journal Entry Set Editor all funds within the set must be balance and the Offset code must always be "NY". No manual inter-fund offsetting or automatic intra-fund offsetting will be allowed.

An additional feature of the Journal Entry process is the capability to update encumbrance information directly from Journal entries. This feature allows the user to create, increase, decrease, disencumber, partially pay, or fully pay encumbrances. The user has the option of updating encumbrance information exclusively, or the option of updating both the General Ledger and the Encumbrance database. This feature could be used to create a new encumbrance against a Purchase Order, encumber additional amounts against a particular Purchase Order, decrease the encumbrance amount on a particular Purchase Order, or disencumber a particular Purchase Order. It could also be used to adjust or record payments made against particular Purchase Orders.

It is important to keep in mind the Balance Type of the Object Codes that are encumbered on the Purchase Order that is being affected by this Journal Entry process. For example, assume that the Object Codes which are encumbered on the Purchase Order are expense type object codes and therefore have a balance type of Debit. To create or increase an encumbrance amount for the Purchase Order, the entries on the Journal Entry screen where the Type field is EN must net to a debit balance. To decrease the encumbrance amount for the Purchase Order, the entries on the Journal Entry screen where the Type field is EN must net to a credit balance. Partial or Full payments must net to a debit balance. These payments are indicated with either a PP or FP respectively in the Type field. Disencumbrances should also net to a debit balance, although it is not as critical because the Purchase Order will be marked as DE regardless of the balance. If the Object Codes on the Purchase Order are not debit balance Object Codes, then the reverse of the preceding example would be true.

Three fields on the Journal Entry screen must be completed to activate this feature. The Secondary Reference field must contain the Purchase Order number. The “Hit EN?” field must contain either a “Y” or an “E”. A “Y” indicates that the entries on this Journal Entry screen are to be distributed to the Encumbrance database AND posted to the General Ledger database. An “E” indicates that the entries on the Journal Entry screen are to be distributed to the Encumbrance database exclusively and are NOT to be posted to the General Ledger database. When using Hit EN = “E”, the Offset Code on all line items for that screen should be set to 'NN'. The Type field indicates the type of encumbrance entry to be made to the Encumbrance database. The valid types are: EN indicating an encumbrance transaction, DE indicating a disencumbrance transaction, PP indicating a partial payment, and FP indicating a final payment. If the Type is set to “EN” and a new encumbrance is being created, some fields will be defaulted on the encumbrance entry: division code will be 'GEN', posting code will be '01', and transaction format will be 'NB'. If alternative defaults for these fields are desired, the Common Code ENGL-DEFAULTS should be set up. See the *Common Code Reference Manual*.

3.2 Standard General Ledger Journal Entries

Many times, a series of transactions are posted to the same accounts, generally for the same amounts, on a regular basis. To make this process easier, create a Master General Journal Entry file from which a copy will be made each month. To create a duplicate copy, use the System Level Software SYUTCP; enter the name of the Master Set File, then the name of the new set file which is to be distributed. Once created, this

new set file may be updated as described in the previous section and distributed to the General Ledger. The Master Set File is then available to be copied again at the next appropriate time. The user is responsible for copying files to destination path. It is recommended to check the destination path before processing this utility. The user should also check the destination path if copying files from DATA directory to POSTED directory or vice-versa. The DATA and POSTED directory contains set files from the many subsystems that created them.

To obtain a Set Proof File Listing of the General Journal Entry, key in GLJEBP. Each Set Proof Listing provides a detail list of the transactions in the same order as entered. It also recalculates the debits and credits to ensure they are balanced. If they are not in balance, a warning is printed.

To distribute the General Journal Entry to the GLDB, key in GLJEDS. The system will prompt for the General Ledger posting date. This will be the posting date used for ALL entries in the set. Options for adjusting the posting process are available. First, the user may indicate if the default Transaction Type is to be changed. The default value is "ST" for standard Journal Entries. Other values are "AC" for Accrual/Reversal entries, "CL" for Closing entries, and "BF" for Balance Forward entries. Second, is the ability to create a Reversal Set of Journal Entries. If the user responds yes, then they will be prompted for a new Set File name and the posting date for the reversing entries. Once the distribution process is activated, all entries in the original set will be placed in the reversing set, with opposite signs. The reversal set may then be posted by the user to the General Ledger on the date desired for reversing entries.

When a set of transactions is distributed to the General Ledger Database, two things take place. First, each individual transaction is associated with the IFAS job number for the specific posting. This provides another audit trail for all transactions which were distributed together. Second, four reports are provided:

1. Set Proof Listing: This should match the listing provided prior to distribution.

2. JE Distribution: A listing of all transactions sorted by account.

3. Journal Entry Postings: A report of all items posted to the GLDB.

4. Audit Report: Provides total Debits and Credits before the distribution, this distribution, and after the distribution.

3.3 Journal Entries Functions

These functions control the creation, editing, and distribution of Journal Entries Set Files.

Which Journal Entries Function would you like to perform? RETURN = UB

AP	JE Approvals Function	BP	JE Set Proof Listing
BX	JE Set Proof (alt. security)	DI	Distribute Inter-Fund JE
DM	Distribute Multi-Date JE Set	DN	Distribute Intra-Fund JE
DP	Directly Post JE Update	DS	Distribute JE Set File
ML	Multi Ledger JE Update	QU	Quick Update (no bal.checks)
RJ	Recurrent JE Function	UB	Create/Update JE Set File

UI	Create/Update Inter-Fund JE	UN	Create/Update Intra-Fund JE
UW	Create/Update JE Wide Format	UX	C/U JE Set (alt. security)
VB	View JE Set as Read Only		

Enter the two-character code, which corresponds, to the Journal Entry function you would like to perform. Choices are described below:

AP	The GLJEAP option goes to JE Approvals Functions menu. See the JE Approval Routing section in this chapter for more information on choices under that menu.
BP	GLJEBP produces a Journal Entry set proof sorted by JEID for the set file entered. No distribution will be performed. The report will list any blocks, warnings, or errors that might keep the set from being distributed. The JE Set Proof Process has the following options available (accessed by right-clicking in Insight):

Special Dollar Formatting	Allows the user to set the format in which the dollar amount fields will be displayed on this report. If this option is selected, the negative number format to use may also be selected.
JEID Page Break	If a page break after every Journal Entry ID is desired, answer "Y". An "N" answer will produce only a line break after each Journal Entry ID.
Show Offsets & Secondary Trns.	Answering "Y" to this question will cause any Primary Offsets or Secondary Transactions to be reported on the Set Proof. If "N" is entered, then these items will be suppressed.
Suppress Budget Checking	Answering "Y" to this question will allow a set that is over budget to post to the GL even if it is a block. You must have database security that allows this or normal budget checking will occur. See the section in the SECURITY guide for GLDB database access for an example of how to set up this security.
Sort within each JEID	Enter "Y" to sort the set file by JE and to change the order of journal entries within each JEID in the set file. Deleted records will be removed from the set. You will also be given the choice to combine like entries within the set. If "N" is entered, the order of the journal entries within the set will not be affected and deleted records will remain in place.

Compressed Version	If an answer of “Y” is given to this question, the “compressed” version of the set proof will be produced. The JL header, contract number, and approval class groups will not be shown when running the set proof in this format. The SORT headings will also be omitted.
Exceptions Only	Enter “Y” to produce an exceptions-only set proof that shows only errors and no detail. Choose "N" to produce a standard set proof.

BX	This function is similar to JE Set Proof Listing (GLJEBP) except that this process will prompt the user to enter a JE special access password. This password (with appropriate security access) will allow the user to run the set proof using an alternate database security access class. See the section on Alternate Security for Journal Entries in this chapter for more information on this function.
DI	Distribute Inter-Fund JE (GLJEDI) functions the same as Distribute JE Set File (GLJEDS) except that only set files created with the inter-fund set editor form (GLJEUI) will be allowed to be distributed and the set proof listing will show additional debit and credit totals by fund.
DM	Distribute Multi-Date JE Set (GLJEDM) functions the same as Distribute JE Set File (GLJEDS) except that the posting date is derived from the posting date field on each JE Set record instead of from the posting date question that is asked at distribution time. When using this mask, if any primary dates are blank on a journal entry in the set, the date from the posting date question asked at run time will used as primary date and the secondary date will pass through unchanged.
DN	Distribute Intra-Fund JE (GLJEDN) functions the same as Distribute JE Set File (GLJEDS) except that only set file created with the intra-fund set editor form (GLJEUN) will be allowed to be distributed and the set proof listing will show additional debit and credit totals by fund.

DP	Directly Post JE Update (GLJEDP) functions the same as Create/Update JE Set File (GLJEUB) but bypasses the normal Set Processing procedures for Journal Entries and immediately posts the entries directly to the General Ledger database. A balanced JEID will be posted directly to the database when the 'Last?' field is set to 'Y'. The user will be prompted if this direct posting is desired. If yes, then the processing will be displayed to the user to verify that the posting completed without error. Once the processing has completed, the user can return to the Journal Entry function or exit the process. The JE direct post process will prompt for a posting date and line printer before requesting a set to create/update. The user may optionally create a reversal set or create the journal entries with a transaction type other than ST (standard transaction). The standard JE form (mode S) will be used as the default form for journal entry set files created using this mask. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.
DS	The JE Set Distribution process will prompt for a posting date and line printer after requesting a set to distribute. The distribution will produce a Journal Entry set proof sorted by JEID including a separate set proof with offsetting entries and secondary transaction for the set file entered. The report will list any blocks, warnings, or errors that might keep the set from being distributed. Set proofs that do not have any blocks or errors will create a JE Distribution Report, JE Postings/Audit Report, and allow the set to be distributed to the GLDB. The JE Set Distribution Process has the following options available (accessed by right-clicking in Insight):

Special Dollar Formatting	Allows the user to set the format in which the dollar amount fields will be displayed on this report. If this option is selected, the negative number format to use may also be selected.
Create a Reversal Set	Enter "Y" to create another JE set to reverse the entries made in the JE set being distributed. If this option is chosen, a new description and reference date may be specified for entries in this new set.
Transaction Type	The transaction type for this set of entries may be changed to AC (Accrual and Reversal transactions), BF (Balance Forward Transactions), CL (Closing Transactions), or ST (Standard Transactions). Normally all Journal Entries are ST type transactions.
JEID Page Break	If a page break after every Journal Entry ID is desired, answer "Y". An "N" answer will produce only a line break after each Journal Entry ID.

Process Expense Allocations	If you do not use the Project Management subsystem to automatically re-allocate expense entries then you can ignore this question. Normally JE sets will not reallocate expense entries even though you use Project Management. This allows you to make adjustments to previously posted items without having to worry about the system generating new entries that you don't want. The purpose of this question is to allow you, the person who distributes the JE set, to have the option to do the re-allocation if desired. If you do want the system to reallocate expenses based on the information in the Project Management subsystem then answer "Y" to this question. Otherwise answer "N".
Suppress Budget Checking	Answering "Y" to this question will allow a set that is over budget to post to the GL even if it is a block. You must have database security that allows this or normal budget checking will occur. See the section in the SECURITY guide for GLDB database access for an example of how to set up this security.
Sort within each JEID	Enter "Y" to sort the set file by JE and to change the order of journal entries within each JEID in the set file. Deleted records will be removed from the set. You will also be given the choice to combine like entries within the set. If "N" is entered, the order of the journal entries within the set will not be affected and deleted records will remain in place.
Compressed Version	If an answer of "Y" is given to this question, the "compressed" version of the set proof will be produced. The JL header, contract number, and approval class groups will not be shown when running the set proof in this format. The SORT headings will also be omitted.
Exceptions Only	Enter "Y" to produce an exceptions-only set proof that shows only errors and no detail. Choose "N" to produce a standard set proof. Additionally the following Common Codes may be setup to notify users when certain conditions occur during the JE set distribution process.
GLEM/EMAIL	Emails an Audit Report to selected users when the GL goes out of balance.
GLJE/EMAIL	Emails a warning message whenever a JE Distribution fails.

ML	For those users who have multiple General Ledgers defined, it may be necessary to create Journal Entries which affect more than one General Ledger per set. This Journal Entry function works exactly like the Create/Update JE Set File (GLJEUB) function with the exception that each line of the Journal Entry can be associated with specific Ledger Codes by placing the two-character General Ledger and Job Ledger Codes in the "GL/JL" column. The lines so specified will post to the appropriate Ledger. Separate posting and audit reports will be produced for each Ledger specified in the set. Of course, the normal balance checking will occur for these entries. The multi-ledger JE form (mode M) will be used as the default form for journal entry set files created using this mask. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.

QU	This function is identical to the Create/Update JE Set File (GLJEUB) except that balance checking is deferred until posting time. The processing of large JE sets may be speeded up through the use of this editor. The standard JE form (mode S) will be used as the default form for journal entry set files created using this mask.
RJ	This menu option goes to Recurrent JE Functions menu. See that Recurrent Journal Entries (RJE) section in this chapter for more information on choices under that menu.
UB	This function creates a new JE set file or allows the user to update an existing one. The standard JE form (mode S) will be used as the default form for journal entry set files created using this mask. A second screen allows free form text to be entered for each JEID. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.
UI	This function is the same as Create/Update JE Set File (GLJEUB) except that Create/Update Inter-Fund JE (GLJEUI) set files must always have an offset code of 'NN' and no manual entries to inter-fund accounts will be allowed. An additional column shows the fund for each JE line item. The Inter-Fund JE form (mode I) will be used as the default form for journal entry set files created using this mask. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.
UN	This function is the same as Create/Update JE Set File (GLJEUB) except that Create/Update Intra-Fund JE (GLJEUN) set files must always have an offset code of 'NY' and no manual entries to inter-fund accounts will be allowed. An additional column shows the fund for each JE line item. The Intra-Fund JE form (mode N) will be used as the default form for journal entry set files created using this mask. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.
UW	Create/Update JE Wide format (GLJEUW) is the same as Create/Update JE Set File (GLJEUB) except that account numbers as large as 40 characters are able to be entered. The Wide-Account JE form (mode W) will be used as the default form for journal entry set files created using this mask. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.
UX	This function is the same as Create/Update JE Set File except that this process will prompt the user to enter a JE special access password. This password (with appropriate security access) will allow the user to create/update the set file using an alternate database security access class. See the Alternate Security for Journal Entries section for more information on this function. The standard JE form (mode S) will be used as the default form for journal entry set files created using this mask. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.
VB	This function is the same as Create/Update JE Set File (GLJEUB) except that the set is opened in read-only mode (meaning that no changes or additions to the set are permitted). This mask will also allow the user to access set files in the posted group without requiring special security setup. See the beginning of this chapter for more information on the Journal Entry Set Editor screens.

3.4 Journal Entry Approval Functions

These functions control the reads, writes, deletes and updates of Journal Entries in the Approval datasets (hereafter known as Approve). Many of these functions are similar to the current Journal Entry functions.

Which JE Approvals Function would you like to perform?

BP	Set Proof JEID in Approve	**BX**	Set Proof JEID (alt. sec.)
DA	Distribute JE Set to Approve	**DI**	Distribute Inter-Fund JE
DM	Dist. Multi-Date to Posted	**DN**	Distribute Intra-Fund JE
DS	Distribute JEID to Posted	**OB**	Set Proof 1-sided JEID
OJ	Aprv/Rej/Update 1-Sided JEID	**OM**	Dist. 1-Sided Multi-Dt to Post
OS	Distribute 1-Sided to Posted	**QU**	A/R/U Quick (no bal. checks)
RA	Report Approved JEIDs	**RN**	Report JEIDs Needing Approval
UB	Aprv/Rej/Update JEID Std Mode	**UI**	Aprv/Rej/Update Inter-Fund JE
UN	Aprv/Rej/Update Intra-Fund JE	**UW**	Aprv/Rej/Update JEID Wide Mode
UX	Aprv/Rej/Update JEID (alt sec)	**VB**	View JEID as Read-Only

Enter the two-character code, which corresponds, to the Journal Entry function you would like to perform. Choices are described below:

BP This function is similar to the standard Journal Entry Set Proof (GLJEBP) except that this process will prompt the user for a single JEID to run the Set Proof on instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

BX This function is similar to the Journal Entry Set Proof using an alternate database security access class (GLJEBX) except that this process will prompt the user for a single JEID to run the Set Proof on instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

DA This process will prompt for a JE Set file name. The contents of the set file (transactions and text) are copied to Approve and then the set file is automatically moved to the posted group. This process supports all JE Set types: Standard, Multi-Ledger, Wide Account, and One-Sided.

DI This function is similar to the Inter-Fund Journal Entry Distribution (GLJEDI) except that this process will prompt the user for a single JEID to Distribute into the GLDB instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

DM This function is similar to the Journal Entry Multi-Date Distribution (GLJEDM) except that this process will prompt the user for a single JEID to Distribute into the GLDB instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

DN This function is similar to the Intra-Fund Journal Entry Distribution (GLJEDN) except that this process will prompt the user for a single JEID to Distribute into the GLDB instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

DS This function is similar to the standard Journal Entry Distribution (GLJEDS) except that this process will prompt the user for a single JEID to Distribute into the GLDB instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

OB This function is similar to the One-Sided Journal Entry Set Proof (GLUTSUOB) except that this process will prompt the user for a single JEID to run the Set Proof on instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

OJ This function is similar to the Create/Update One-Sided Journal Entry Set File (GLUTSUOJ) except that this process will prompt the user for a single JEID to Approve, Reject, or Update instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. See the description for the UB option for more information on editing JEIDs in approve.

OM This function is similar to the One-Sided Journal Entry Multi-Date Distribution (GLUTSUOM) except that this process will prompt the user for a single JEID to Distribute into the GLDB instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

OS This function is similar to the One-Sided Journal Entry Distribution (GLUTSUOS) except that this process will prompt the user for a single JEID to Distribute into the GLDB instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA).

QU This function is similar to the Quick Create/Update Journal Entry Set File (GLJEQU) that defers balance checking until posting time except that this process will prompt the user for a single JEID to Approve, Reject, or Update instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. See the description for the UB option for more information on editing JEIDs in approve.

RA This report shows all approvals that have been completed that match the entered selection criteria. Selection may be made on the following criteria: JEID, Approver User ID, and approval 'As of' Date.

RN This mask runs a report of JEIDs needing approvals. The report supports selection criteria by JEID, Approver User ID, 'As of' Date, and full account and transaction level selection criteria. The report supports the following sort orders: 1) JEID, 2) JE-ID within Approver User ID, and 3) JEID within JEID Approval Class Group. There are three levels of verbosity: 1) The short format shows a summary record for each JEID that lists the first four approvers needed to approve this JEID. 2) The normal format shows partial detail for each transaction and the first two approvers needed to approve the JEID and Transaction level approvals for this JEID. 3) The verbose format shows all transaction detail along with the first two approvers needed to approve the JEID and Transaction level approvals for this JEID.

UB This function is similar to the Create/Update Journal Entry Set File (GLJEUB) except that this process will prompt the user for a single JEID to Approve, Reject, or Update. This JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. The Set Editor will begin in Approve Mode. The first position of the JEID and Transaction level Aprv fields will identify the approval condition for the JEID or Transaction level approval class group (JACG or TACG). The field will be blank if all approvals required for the JACG or TACG codes have been approved (or if there are none). The field will contain an asterisk (*) if

JACG or TACG codes still need to be approved but the user is not authorized to approve any of them. The approval condition field will contain a "1" if any JACG or TACG codes still need to be approved for which the user is authorized to approve as a primary approver. The field will contain a "2" if any JACG or TACG codes still need to be approved for which the user is authorized to approve as a secondary approver and no primary ones exist. The user The user can replace the "1" with an "A" to approver primaries or replace the "2" with a "S" to approve secondaries. A "R" may be placed in this field to reject all prior approvals for this JACG or TACG. Rejecting it removes all JACG or TACG approvals for all approval codes for that specific transaction. If the "Reject to Disk" special user capability flag is set to "Y" for the user who created this JEID, all records for this JEID will be rejected completely out of the approve area and into a JE set file in the DATA group. Except for approvals, this Set Editor functions like GLJEUB. See Chapter 3 for more information.

UI This function is similar to the Intra-Fund Create/Update Journal Entry Set File (GLJEUI) except that this process will prompt the user for a single JEID to Approve, Reject, or Update instead of a file. The JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. See the description for the UB option for more information on editing JEIDs in approve.

UN This function is similar to the Intra-Fund Create/Update Journal Entry Set File (GLJEUN) except that this process will prompt the user for a single JEID to Approve, Reject, or Update instead of a file. The JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. See the description for the UB option for more information on editing JEIDs in approve.

UW This function is similar to the Wide Account Create/Update Journal Entry Set File (GLJEUW) except that this process will prompt the user for a single JEID to Approve, Reject, or Update instead of a file. A Wide Account, for purposes of this process, is any account string longer than 37 characters. The JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. See the description for the UB option for more information on editing JEIDs in approve.

UX This function is similar to the Create/Update Journal Entry Set File using and alternate database security access class (GLJEUX) except that this process will prompt the user for a single JEID to Approve, Reject, or Update instead of a file. This JEID must have previously been distributed to Approve (using GLJEAPDA). If the set is modified, all approvals for this JEID will be cleared. See the description for the UB option for more information on editing JEIDs in approve.

VB This function is similar to the View JE Set as Read-Only (GLJEVB) process except that it will prompt the user for a single JEID to View interactively through the JE Set Editor Screens. This JEID must have previously been distributed to Approve (using GLJEAPDA). The set may not be approved or modified when using this mask.

3.5 Recurring Calculation Process

The Recurring Calc process is under development and scheduled for release in June 2008.

3.6 General Transaction Processing

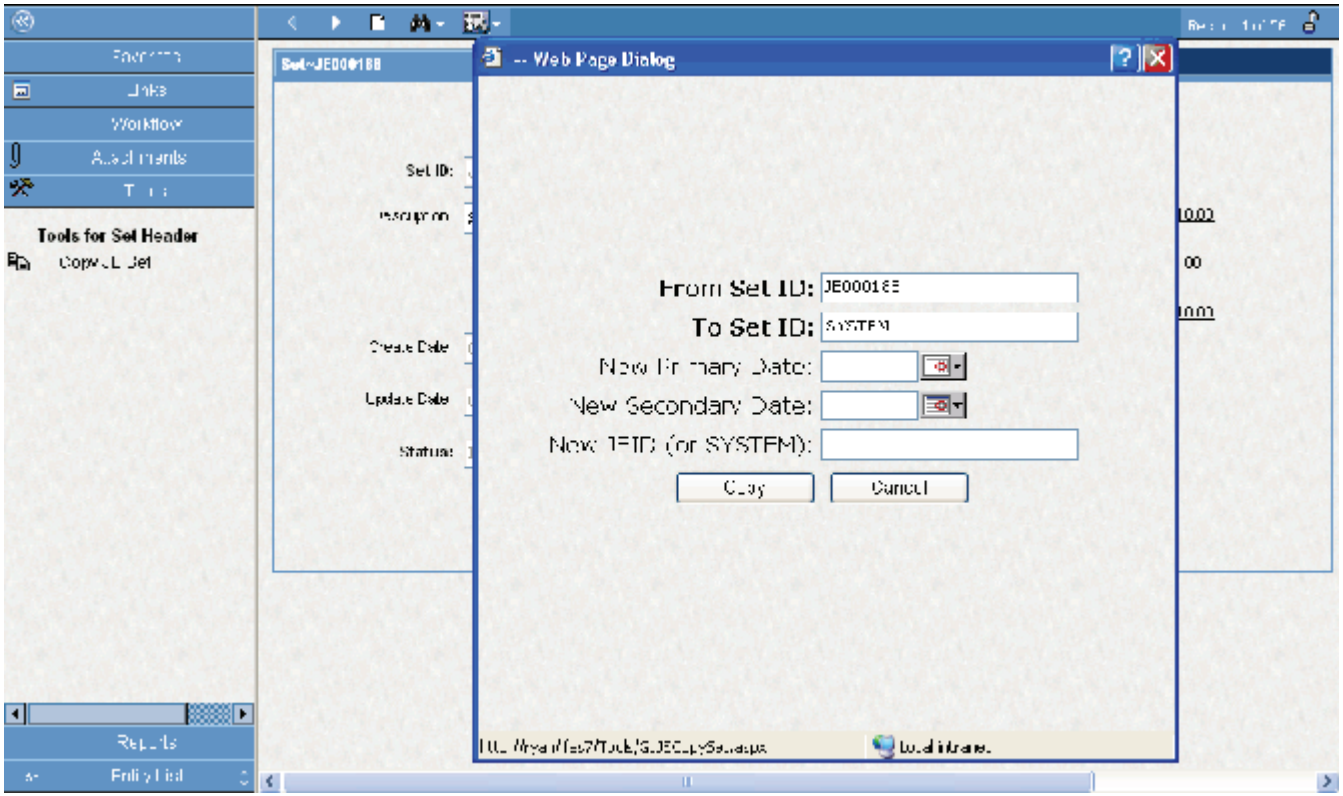
All transactions within the General Ledger contain detailed information relating to the source, description, and audit trail necessary for reconciliation of balances. These detailed characteristics are unique for each subsystem from which they are posted. The following lists all data which may be coded to a single transaction.

GL Organization Key
GL Object Code
JL Organization Key (optional)
JL Object Code (optional)
Transaction dollar amount
Transaction unit amount (optional)
Miscellaneous Code
Work Order number (optional)
Primary Reference Journal Entries User-defined JE ID#
Accounts Payable Vendor Invoice #
Accounts Receivable Customer Invoice #
Cash Receipt Receipt #
Fixed Assets Asset ID#
Payroll Pay Period Number
Primary Date Accounting Period Posting Date
Secondary Reference Journal Entries User-defined (optional)
Accounts Payable Purchase Order #
Accounts Receivable User-defined (optional)
Cash Receipt Invoice # or Bank Slip #
Fixed Assets Posting Code defined
Payroll Posting Code defined
Secondary Date of Journal Entries User-defined(optional)
All Others Posting Date
Person/Entity ID Vendor ID# from A/P - Customer ID# from A/R
A/P Check Number Created by Accounts Payable
Description of Journal Entries User-defined
Accounts Payable Multiple User formats
Accounts Receivable Multiple User formats
Cash Receipt User-defined
Fixed Assets User-defined
Payroll User-defined

Units of Measure	System Generated from A/P and A/R
Contract #	User-defined (optional)
Accounting Period	System Generated
Transaction Type	System Generated
PEDB Code	Name/Address Database in which ID is found
Set ID #	User-defined
IFAS Job #	System generated
User ID	System generated
Subsystem ID	JE, AP, AR, CR, FA, PY

3.7 Copy Set Tool for GLJEUP

A Copy JE Set Tool is available under the Tools section of the Options Bar. The user must have access to the “COPY” command in NUUPUS use this tool. The Copy JE Set tool allows you to copy an existing JE set (and corresponding detail records) from one Set ID to another. The Set copied from will remain unchanged.



By default, the current Set ID will appear in the “From Set ID” field. This may be changed if you wish to copy from another Set ID. By default, “SYSTEM” will appear in the “To Set ID” field. The “SYSTEM” in the “To Set ID” will generate a new Set ID from the BTCHIDJE seed value. If desired, a user-entered “Set ID” may also be used. Set Id naming rules apply.

New Primary and Secondary JE Dates may optionally be provided otherwise the dates on each detail record will be copied over as is.

Specifying a new JEID is recommended. By default, the original JEID will be used. Optionally “SYSTEM” or “SYSTEMxx” may be entered to generate a new JEID from the JEID or JEID.xx seed value.

If the message “Copy Complete” comes up at the top of the screen, the copy was successful and you will be redirected to the new Set ID. If the copy failed, the reason for the failure will be displayed at the top of the screen and the Copy Set window will remain open until you complete a successful copy or hit the cancel button.

4 Reporting

Refer to the CDD guide for details about the reporting process. You will find examples of GL reports in the Sample Reports Notebook.

5 : Fiscal Year-End

5.1 Fiscal Year-End Procedures

This section of the User Guide will provide a discussion of the processes used to prepare the General Ledger for the beginning of a new fiscal year and different methods to affect the old fiscal year transactions. The user should be familiar with the Recurring Calculation process before continuing. Prior to preparing for the year-end process, it should have been determined how closing entries will be maintained within the system. Will they be posted as 'ST' Standard or as 'CL' Closing entries? In closing one fiscal period and beginning a new one, two steps are involved:

1. Posting beginning balances in the new fiscal year.
2. Determining the level of detail for which prior year transactions will be maintained in the General Ledger.

5.2 New Year Balances

Beginning new year balances are calculated through the use of a Recurring Calculation; which analyzes all of the year's transactions and creates an entry summarizing them at whatever level is specified in the calculation. These entries may be posted from the individual detail transaction level to any control total level. Also, any subset of the General Ledger may be closed concurrently with other areas or separately, depending on the client's requirements. Specific accrual and reversal entries may be posted at the same time to the appropriate period or separately. The user determines the timing of posting all period closing entries. Special closing entries may be separated from balance forward entries and/or audit required adjustments. Review of the Recurring Calculation function in this User Guide may be helpful before proceeding.

As the examples outline, different posting strategies may be used for different fund and/or organization key GL structures. Generally, there should be a one-to-one correspondence between each asset and liability account balance from the old year end to the new year beginning. For fund balance and intrafund accounts, it may be helpful to combine the activity for a range of Org. Keys and Object Codes. A key concept within IFAS is that it is not necessary to physically close revenue and expenditure accounts into fund balance. The system is sensitive to the defined fiscal year and therefore, all revenue and expenditures automatically begin with zero balances, unless specific accrual/reversal entries are posted. The creation of any beginning fund balance in the new fiscal year is prepared by the calculation of the ending fund balance plus all appropriate revenue/additions minus all appropriate expenditure/deductions. The calculated result is then posted into the new year. This process allows the user to easily provide multiple fiscal year comparison reporting since revenue and expenditures are not closed. Of course, an alternative approach is to define a specific closing period in the definition of the Ledger and post the entries to close Revenue and Expense to Fund Balance. Next, we would create another Recurring Calculation to prepare all the balance forward entries into the new fiscal year.

For purposes of our examples, the Chart of Accounts defined at the beginning of this guide will be used. The following strategies will be used for each fund:

5.2.1 General Fund

Assets, Liabilities, and Fund Balance at the Fund Admin. level and Income and Expense at the Organization Key level.

5.2.2 Capital Improvements Fund

Assets & Liabilities at the Fund Admin. level and Fund Balance, Income and Expense at the Organization Key level.

5.2.3 Grants and Contracts Fund

Individual Organization Keys having Assets, Liabilities, Fund Balance, Income and Expense.

5.2.4 Pooled Cash Fund

Assets only.

5.3 Prior Year Transactions

After the audit has been completed and beginning balances have been distributed in the new fiscal year, it may be necessary to decide what to do with prior year transactions. There are three choices:

1. Do nothing. All detail transactions will be available for reporting and inquiry. However, this may not be compatible with existing disc storage capacity.

2. Store these transactions to some other medium such as tape or microfiche and purge them all from the General Ledger database. However, this will not provide the capability to produce comparative year reports.

3. Store the transactions as in (2) and then collapse all detail on a monthly basis to a single number for each account key and object by use of the GL balance forward utility, **GLUTSUCL**. Although the ability to access detailed transactions on the system is lost, comparative reports may still be produced. As an alternative, the user may wish to collapse detail for only some parts of the GL (i.e., Asset, Liability, Fund Balance and Intrafund object codes). This would keep the detail for all Revenue and Expenditure accounts.

As a greater number of years of data are maintained, different strategies for various time periods may be identified. For instance, last year's data through three years previous would be collapsed on a monthly basis. All other prior years would be maintained as year-end balances only. In addition, there may be different strategies for separate sections of the Chart of Accounts. Debt Service or Plant Funds may maintain detail transactions for many years, while other funds only have monthly balances. Selection Criteria is provided to allow specific sections of the Chart of Accounts to be collapsed as desired.

6 GL Interface

6.1 Overview

Good Fund Accounting practices would include three particular levels of accounting. The lowest level is at the department/cost center level where we find an Org. Key; which represents some combination of fund, function, division and budget officer. The next level is the fund level; where we find an Org. Key represents a particular fund. The highest level is where an Org. Key is set up to hold all bank accounts and common investment pools.

The foundation of Fund Accounting is to provide a way to keep track of the relationships among these three levels of accounting. The relationship between the lowest level and the fund level is modeled by a special Object Code called 'Intra-fund Clearing;' or a similar title. This special Object Code is associated with the Org. Keys which represent the departments or cost centers and is associated with the Org. Key which represents the fund. As Accounts Payable, Accounts Receivable, Payroll and other activity are recorded for the department or cost center, the 'Intra-fund Clearing' account is posted with control totals at the department or cost center and with corresponding, opposite signed, entries at the fund level. The relationship between each fund level and the cash administration level is modeled by a special Object Code called Inter-fund; 'Due To/From Other Funds', 'Claim on Cash', 'Cash in Bank, Clearing' or a similar title. This special Object Code is associated with each fund and is associated with the Org. Key which represents the cash administration. As Accounts Payable and Payroll checks are written and as Cash Receipts are recorded, this 'Claim on Cash' account is posted with control totals at the fund level and with corresponding, opposite signed, entries at the cash administration level.

As we have discussed, three levels of accounting are required: The department level, the fund level and the entity; administration level. Thus, in IFAS, we typically find a unique Org. Key set up to represent each department, and fund, with a single Org. Key set up to represent entity administration. We will refer to these different levels as 'Department Org. Keys', 'Fund Org. Keys' and 'Entity Org. Keys'. To demonstrate the flow of transactions required for proper fund accounting among these three different levels, the following example of Accounts Payable activity is presented.

For the example below, assume that the following Org. Key definitions exist:

Org. Key	**Definition**
0000	Cash Administration
0010	Fund 10 Administration
0020	Fund 20 Administration
1000	Dept One of Fund 10
2000	Dept Two of Fund 10
3000	Dept Three of Fund 20

Object Code	**Description**
1021	Cash in Bank
1025	Intra-fund Clearing
1029	Claim on Cash
3500	Accounts Payable
8000	Expense Type One
8500	Expense Type Two

Org. Key	**Object Code**	**Amount**
1000	8000	$10
1000	8500	$20
2000	8500	$40

3000	8000	$90

When expenses are posted, the following entries would be made (Definitions & Descriptions have been abbreviated for reference purposes):

Dept 1 - Expense 1 (1000-8000)	10	
Dept 1 - Expense 2 (1000-8500)	20	
Dept 1 - Intra Clearing (1000-1025)		30
Dept 2 - Expense 2 (2000-8500)	40	
Dept 2 - Intra Clearing (2000-1025)		40
Fund 10 Admin –Intra Clear(0010-1025)	70	
Fund 10 Admin – AP (0010-3500)		70
Dept 3 – Expense 1 (3000-8000)	90	
Dept 3 – Intra Clearing (3000-1025)		90
Fund 20 Admin-Intra Clear(0020-1025)	90	
Fund 20 Admin-AP (0020-3500)		90

Then, when checks are written, we would see these entries:

Fund 10 Admin – AP (0010-3500)	70	
Fund 10 Admin – COC (0010-1029)		70
Fund 20 Admin – AP (0020-3500)	90	
Fund 20 Admin – COC (0020-1029)		90
Cash Admin – COC (0000-1029)	160	
Cash Admin – Cash (0000-1021)		160

In the above example, we can see that Object Code 1025 is used to represent the relationship between departments and funds and Object Code 1029 is used to represent the relationship of funds to cash administration. Since many not-for-profit organizations have different organizational

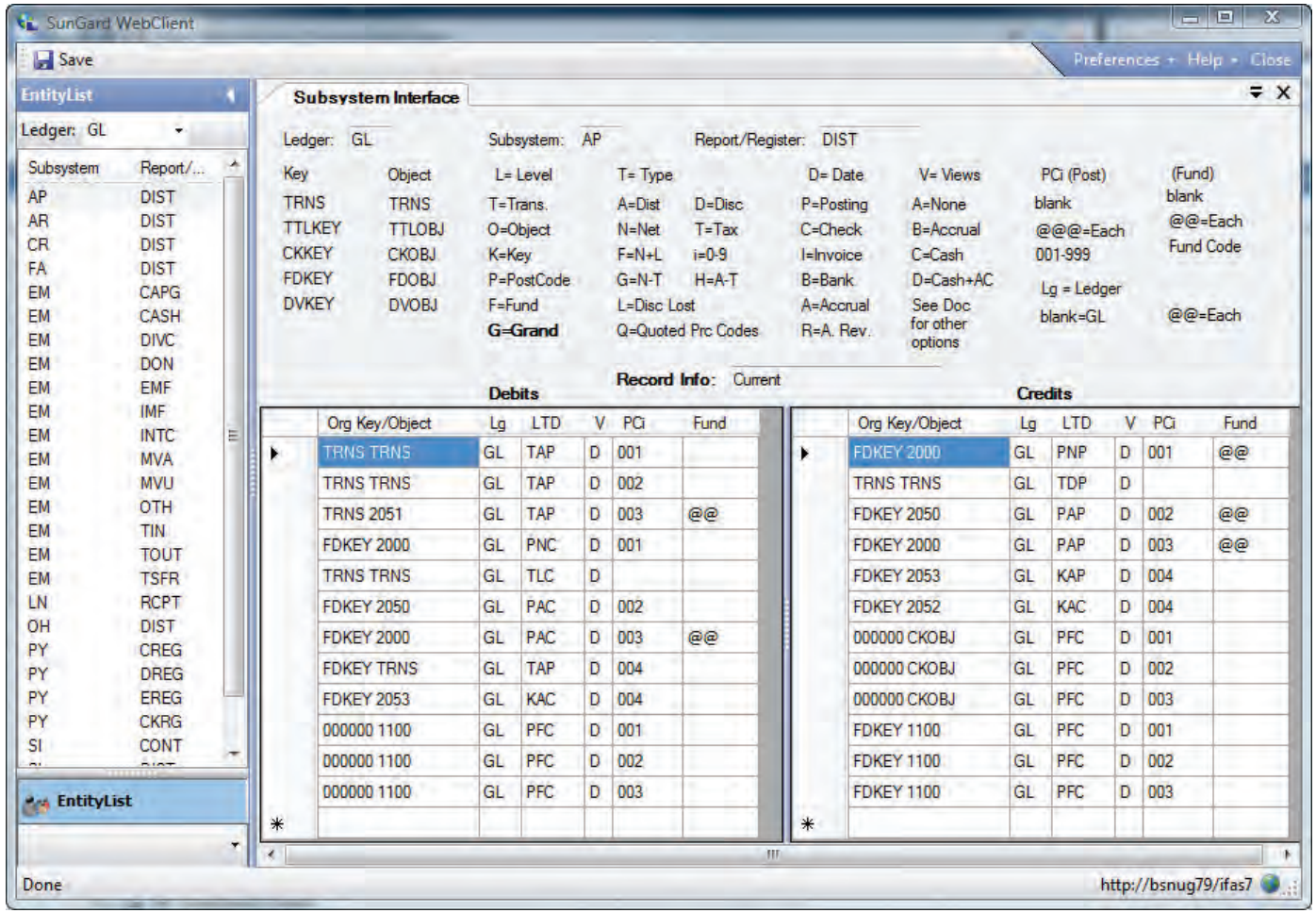
parts and since there exist many different acceptable posting methods, IFAS has been designed to allow the organization to define any organizational parts and to allow the accountant to define any type of automatic posting.

Before presenting the details of posting definitions, it may be helpful to outline the relationship of various coding structures to the data entry and set file distribution process. The following diagram depicts these relationships. When the user is in the Create/Update Set File function, a Posting Code is required to be able to direct the specific logic for all balancing entries. The Posting Preferences process derives the Posting Code based on user defined logic for each subsystem.

When distributing the Set File, the Posting Code directs the software to use a defined strategy within the GL SUBSYSTEM INTERFACE. This strategy identifies the balancing entries required for double entry accounting, fund level control accounts, cash accounts, or other types of special entries which may be required. These entries become a part of the Post File along with the original Set File records and are posted to the General Ledger.

6.2 GL Subsystem Interface Definition

All General Ledger (GL) posting from all subsidiary systems is defined on the GL SUBSYSTEM INTERFACE screen shown below. To display this screen key in **GLUTSPSI**.



This screen is used to define those entries which are to be posted to the General Ledger as the result of transaction activity in a given subsidiary system (A/P, A/R, Cash Receipts, Payroll, Stores Inventory). The posting of subsidiary system activity is directed on an individual transaction basis or at a summary level (by object, by department, by posting code, by fund code, etc.).

The process of defining a General Ledger interface from a subsidiary system is one in which the user specifies, by subsidiary system, how this posting is to occur. Thus, the GL SUBSYSTEM INTERFACE screen is the form on which one says: “In the GL account, post this value for

this type of A/R (or A/P or PY) transaction." The remainder of this section will describe the different ways in which one may identify a GL account and a particular value.

The top of the screen contains the identification of the Ledger, Subsystem, and the Report/Register which prints the values that are to be posted. If more entries need to be defined than will fit on one screen, simply press the ENTER key when the first screen is filled and continue when the screen is blanked out, indicating that the first screen full of entries has been written to the database.

The 'Ledger' field identifies the specific GL Ledger for which the Subsystem Interface is being defined. In this manner, each defined ledger may have a different set of posting definitions. This code should be identical to the two-character code defined within the General Ledger General Information, **GLUPGN**. (Note: The Payroll Subsystem Interface always uses 'GL'. Please see the Payroll subsection below for additional information.)

The 'Subsystem and Report/Register' fields are used to group transactions into categories within subsidiary systems. The valid codes which may be entered are as follows:

Subsystem	Report/Register	Meaning
AP	DIST	Accounts Payable Values
AR	DIST	Accounts Receivable Values
CR	DIST	Cash Receipts Values
FA	DIST	Fixed Asset Values
LN	RCPT	Loan Cash Receipt Values
OH	DIST	Special Open Hold AP Values
PO	DIST	Purchasing/Receiving (Client Specific) Values
PY	DREG	Payroll Deduction Reg. Values
PY	CREG	Payroll Contribution Reg. Values
PY	EREG	Payroll Earnings Register Values
PY	CKRG	Payroll Check Register Values
SI	DIST	Stores Inventory Values
SI	RCVG	SI Set Receiving
SI	CONT	Inventory Count Files

The values shown in the above chart are to be placed into the fields labeled 'Subsystem and Report/Register' on the top of the form. If BROWSE-mode is entered, use the 'Find' fields at the bottom of the form to cause browsing to continue with an entered 'Subsystem and

Report/Register'. Browsing of a particular Subsystem and Report/Register always starts with the first screen full of entries; if more screens exist for a given interface description.

As stated, this form is used to specify which GL accounts are to be posted with which values. The reader will note that the main portion of the screen is divided into DEBIT and CREDIT sides: Fill in the left half for all accounts which are to be debited; fill in the right half for all accounts which are to be credited. If the system computed value is negative, the system will switch a debit to a credit. Each side, debit and credit, contains eight lines each; one line is used to define each desired GL posting. You first specify which GL account is to be posted (Org. Key and Object); and then indicate which value is to be posted (L, T, D, V, PCi, and FD). A general description of each possible GL account identification will be provided; then, a description of how a particular value may be specified.

HOW TO IDENTIFY THE GL ACCOUNT: The GL account identification consists of identifying the two parts which are used to post GL transactions. The first part is called the department, cost center, project, key or some organization ID (we call it the 'Org. Key') and the second is the Object Code. Several codes have been established to make the definition of the GL interface easier. The following describes all possible codes for directing entries from subsidiary systems (A/R, A/P, etc.) to the proper GL account. Please note that not all of these will make sense for a particular 'Subsystem and Report/Register' and some may never be used by a given client.

ORGANIZATION KEY ENTRY (Org. Key)

Org. Key	Description
TRNS	Use the Org. Key entered on the transaction. This is used when the Chief Accountant would like to see individual transaction postings in the GL.
TTLKEY	Total all the entries within the set for the same Org. Key and post a single summary transaction.
CKKEY	Certain types of transactions have a Check Stock ID code entered as part of the transaction (i.e., Cash Receipts), check writing from Accounts Payable and Payroll. These Check Stock ID's can be associated with a particular Org. Key in the Common Codes File. (This process is described later in this section). When the user places CKKEY in this field, the system automatically retrieves the Org. Key associated with the given Check Stock ID.
FDKEY	Using the Common Codes File posting of inter-fund (and other) entries at the fund level may be effected by directing posting to a single Org. Key for each fund.
CKFDKEY	A combination of FDKEY and CKKEY. It is the Fund Administration Key (FDID) within the Fund identified by the Org. Key defined on the CKID Common Code.
DVKEY	From the Subsystem Division entered in the set, use the Org. Key entered on the Division Definition Form (supported only by AR at this time).
xxxxxxxxxx	Use the specific Org. Key entered on this form.

OBJECT CODE ENTRY (Object)

Object	Description
TRNS	Use the Object Code entered on the transaction. This is used when the Chief Accountant would like to see individual t ransaction postings in the GL.
TTLOBJ	Total all the entries within the set made to the same object code and post a single summary transaction.
CKOBJ	Certain types of transactions have a Check Stock ID code entered as part of the transaction (e.g., Cash Receipts, Accounts Payable, and Payroll). These Check Stock ID's can be associated with a particular Object Code in the Common Codes File (this process is described later in this section). When the user places CKOBJ in this field, the system automatically retrieves the Object Code associated with the given Check Stock ID.
FDOBJ	When detail transactions are entered, a particular Org. Key and Object Code is specified. Each Org. Key is associated with a particular Fund Code. These Fund Codes can be associated with a particular Object Code in the Common Codes File (this process is described later in this section). When the user places FDOBJ in this field, the system automatically retrieves the Object Code associated with the given Fund Code.
CKFDOBJ	A combination of FDOBJ and CKKEY. It is the Fund Administration (FDID) Object Code within the Fund identified by the Org. Key defined on the CKID Common Code.
DVOBJ	From the Subsystem Division entered in the set, use the Object Code entered on the Division Definition Form (supported only by AR at this time).
xxxxxxxx	Use the Object Code entered on this form.

LEVEL (L)

L	Description
T	Individual transaction level. This is only used when the Org Key is 'TRNS' or 'xxxxxxxxxx' and the Object is 'TRNS' or 'xxxxxxxx'.
O	Object Code total level.
K	Org. Key total level.
P	Posting Code total level.
F	Fund total level.
G	Grand total, all transactions.
X	Pay Type Total.
Z	Finance Code Total.

Any control total postings for O, K, P, F, and G will be posted to the General Ledger with a transaction description as follows:

AUTO ID:BP004681 JB:000623

'AUTO' indicates that this is an automatic total posting.

'ID' is followed by the Set ID Number for all entries in the transaction set.

'JB' is followed by the IFAS Job Number which is automatically assigned at the time of posting.

Any control total postings for X will be posted to the General Ledger with a transaction description as follows:

AUTO Pay: _____ JB:_____

'AUTO' indicates that this is an automatic total posting.

'PAY' is followed by the Pay Type Code for all entries in the transaction set.

'JB' is followed by the IFAS Job Number which is automatically assigned at the time of posting.

Any control total postings for Z will be posted to the General Ledger with a transaction description as follows:

AUTO Fin: _____ JB:_____

'AUTO' indicates that this is an automatic total posting.

'FIN' is followed by the AR Finance Code for all entries in the transaction set.

'JB' is followed by the IFAS Job Number which is automatically assigned at the time of posting.

TYPE (T)

T	Description
=	==========
A	Distribution amount which is Qty x Price + Tax + Charges + Duty (A+T+C+U).
D	Discount amount (D).
N	Net amount which is Qty x Price - Discount + Tax + Charges + Duty (A-D+T+C+U).
T	Tax Amounts (T).
F	Net plus Discounts Lost.
F.	Net Amount plus Discounts Lost (N - L).
G.	Net Amount less Tax (A-D+C+U).
H.	Distribution Amount less Tax (A+C+U).

I Digit 0, 1, 2, ... 9. Used primarily with Payroll subsystem postings.
L Discounts Lost (L).
C. Charges (C).
U. Duty (U).
See Section 7-G for optional system type definitions.

DATE (D)

D Description
= ==========

P Posting date.
C Check date.
I Invoice date.
B Bank date.
A Accrual Entry (used only if posting both Cash **and** Accrual Balances).
R Accrual Reversal (used only if posting both Cash **and** Accrual Balances).

VIEWS (V)

V Description
= ==========

A None
B Accrual Only
C Cash Only
D Accrual and Cash
E Modified Only
F Accrual an Modified
G Cash and Modified
H Accrual, Cash, Modified
I User-Custom Only
J Accrual and User-Custom
K Cash and User-Custom
L Accrual, Cash, and User-Custom
M Modified and User-Custom

N	Accrual, Modified, and User-Custom
O	Cash, Modified, and User-Custom
P	Accrual, Cash, Modified, and User-Custom
Q	Derive from Part Misc Flag 1 for Fund
R	Derive from Assoc. Value 1 of FDID Code

Under almost all instances, <u>D</u> will be used.

Posting Codes (PCi)

PC Description

=== ==========

Blank implies that the posting is not related to a specific posting code.

@@@ Each posting code.

iii Posting code 000-999. Posting Codes are used to direct the IFAS posting logic to create different automatic entries. Each code is associated with an Org. Key and Object Code to tell the system which fully qualified account to use in posting transactions to the GL. This is important when the client has defined multiple accounts in the GL which should receive postings from the various subsystems (i.e., Accounts Payable, Accounts Receivable, Payroll, etc.). At the point of initial data entry, within a subsystem the client will indicate the specific posting code to be used for each transaction entered. When entries are distributed to the GL, the detail transactions will be summarized at the LEVEL (L) defined for the posting code used and posted to the defined Org. Key and Object Code. Where multiple accounts are not necessary, a specific posting does not need to be defined. In the examples developed below, effective use of posting codes is discussed under varying situations.

Fund Description

== ==========

Blank implies that this is not a fund level posting.

@@ Each Fund.

ii Fund code.

EXAMPLES

Before developing examples for each of the subsystems, a very simple list of Object Codes and GL Accounts will be defined. Let's assume the following Object Codes exist:

Object	Description
====	==========
1021	Cash in bank, A/P; Stock ID: 'A'
1022	Cash in bank, A/P; Stock ID: 'B'
1023	Cash in bank, Payroll; Stock ID: 'P'
1025	Intra-fund Clearing Account
1029	Claim on Cash
2000	Fixed Assets
3001	Accounts Receivable, Type One
3002	Accounts Receivable, Type Two
4000	Accounts Payable, General
4010	Receiving Clearing
4501	Employer Fringe Liability, Type One
4502	Employer Fringe Liability, Type Two
4601	Employee Deduction Liability, Type One
4602	Employee Deduction Liability, Type Two
4900	Depreciation
5000	Fund Balance
6000	Discounts Earned
6001	Revenue, Type One
6002	Revenue, Type Two
6900	Gain or Loss on Sale of Assets
8001	Expenditure, Salaries and Wages
8501	Employer Fringe Expense, Type One
8502	Employer Fringe Expense, Type Two
8801	General Expense, Type One
8802	General Expense, Type Two
8900	Depreciation Expense

Of particular importance are the Object Codes 1025 and 1029. We use 1025 to show the 'intra-fund' transfers between the central administrative Org. Keys and the Cost Center Org. Keys to which they relate. We use 1029 to show the 'inter-fund' relationships between the central administrative Org. Keys and the Org. Key used to hold the bank accounts.

Recall from the user guide documentation, each set of organizational parts is uniquely identified by an Org. Key. Any transaction posted to the General Ledger must have two parts: An Organizational Part and an Object Code part. Above, sample Object Codes were provided which are to be used in the developed examples. Now some Org. Keys will be defined which will be used to specify the Organizational Part of the transactions which are to be posted to the General Ledger.

It is typical that a client will create Org. Keys for three major types of organizational designations. The first type represents the Department Org. Keys: These are Org. Keys which identify cost centers, projects, basic budget units, departments, and other low level organizational units. These types of Org. Keys will generally only be associated with revenue and expenditure activity. The second type represents the Fund Org. Keys. These are Org. Keys which identify a fund globally. They typically hold assets, liabilities, and fund balances. The third type of Org.

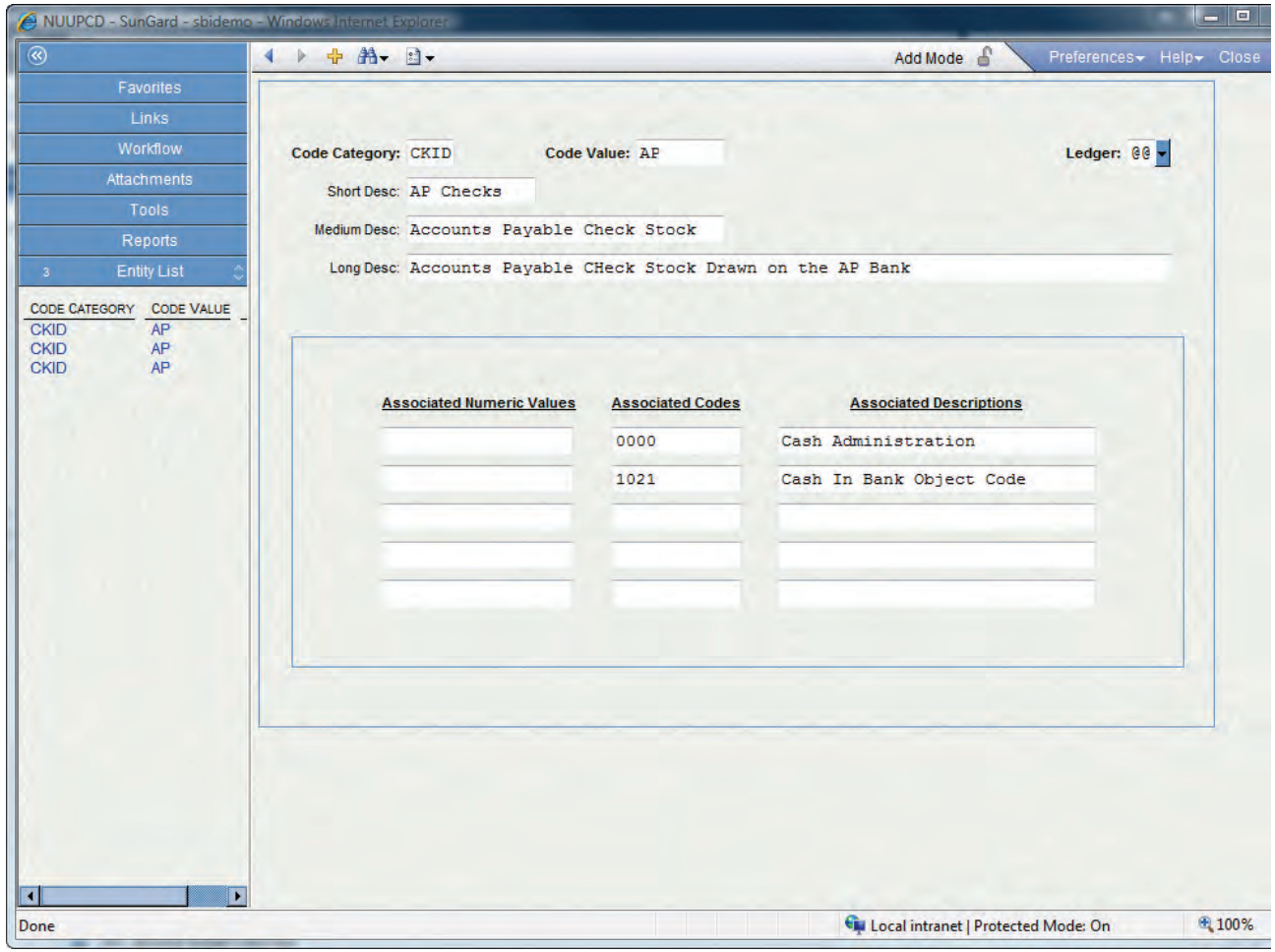
Keys represents the Cash Org. Keys. These are Org. Keys which hold all bank accounts and may hold pooled investments which cross fund boundaries. Generally, only one such Org. Key exists for the entire organization.

Org. Key	Fund	Description
0000	NA	Account used for bank accounts
0010	10	General Fund 10 Administration
0020	20	Restricted Fund 20 Administration
0030	30	Restricted Fund 30 Administration
1000	10	Department One
2000	10	Department Two
3000	20	Department Three
4000	20	Department Four
5000	30	Department Five
6000	30	Department Six

In the above list, we will assume that all the bank accounts are held in Org. Key '0000', a generic Org. Key which is not associated with a significant Fund ID.

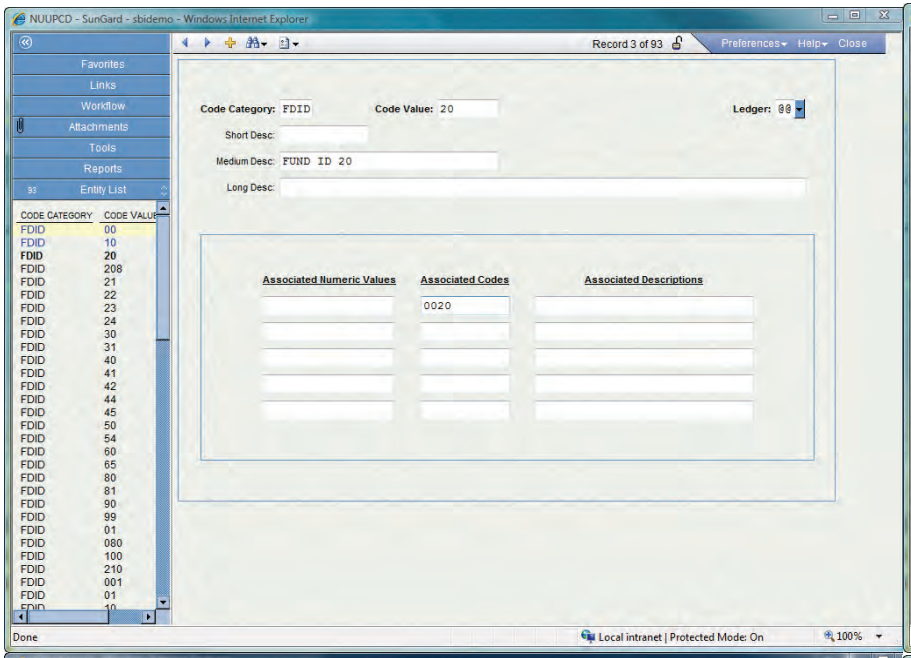
Note that 'CKKEY' and 'FDKEY' may be specified for the Org. Key which is to be posted. This means that when information is entered on the GL SUBSYSTEM INTERFACE screen under GLUTSPSI, we associated a particular Org. Key for a given Check Stock ID (CKKEY) or for a given Fund ID (FDKEY). Also note that 'CKOBJ' or 'FDOBJ' may be specified for the Object Code which is to be posted. Once again, from the GL SUBSYSTEM INTERFACE screen, a particular Object Code for a given Check Stock ID (CKOBJ) or for a given Fund ID (FDOBJ) is associated. Therefore, before developing the specific examples of some typical GL Subsystem Interfaces, the following information will describe how Org. Keys and Object Codes are associated with particular Check Stock ID's and Fund ID's.

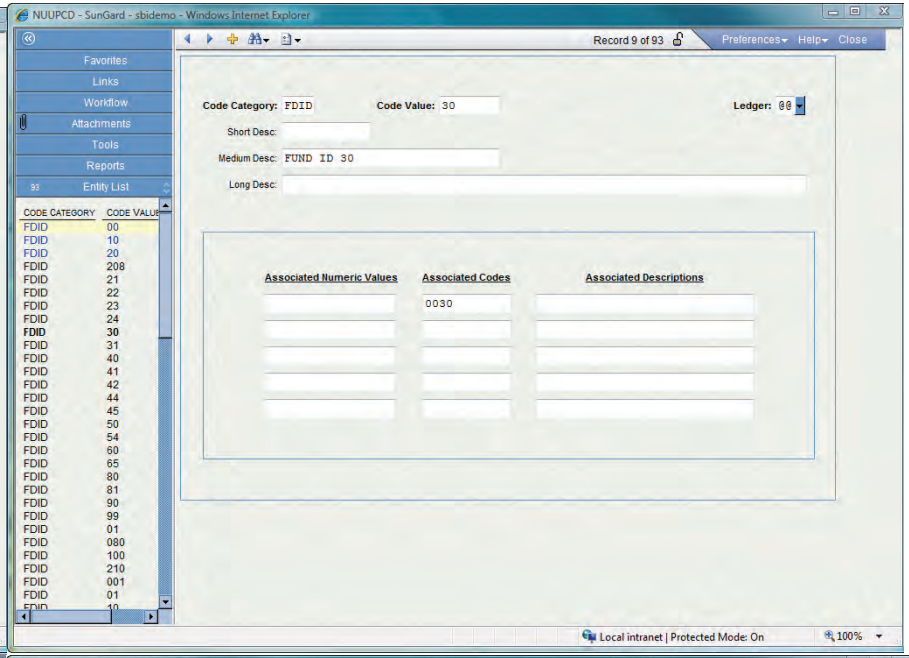
If the client is going to use 'CKKEY', 'CKOBJ', 'FDKEY' or 'FDOBJ' (most will), then the client must make certain that each Check Stock ID and each Fund ID is associated with a given Org. Key and/or Object Code for purposes of GL posting. This is done in the Common Codes part of the NUCLEUS system. The client keys in **NUUPCD** to access the form on which these codes may be entered. The form appears as on the following page.

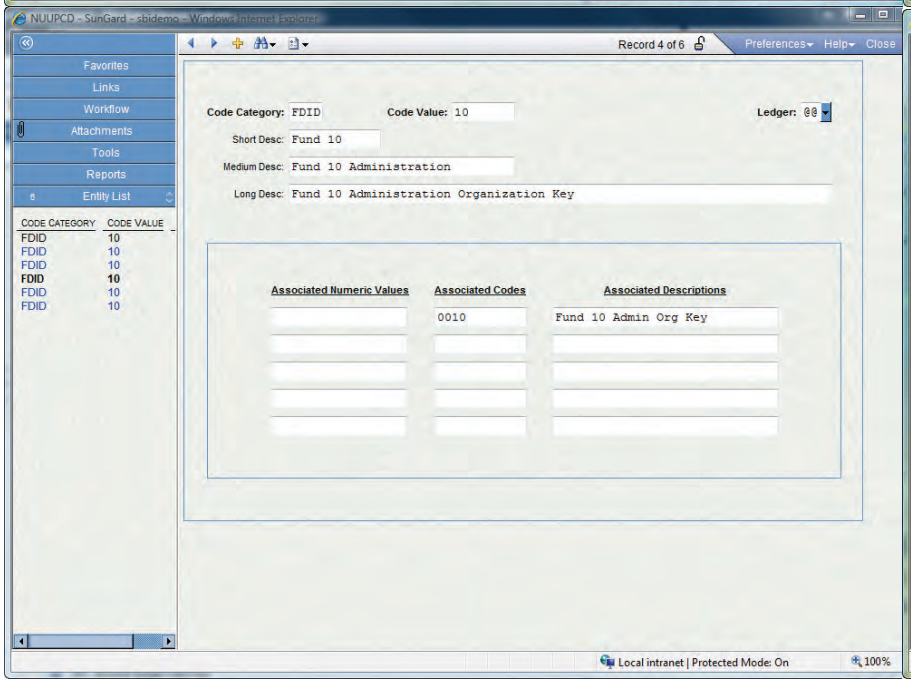


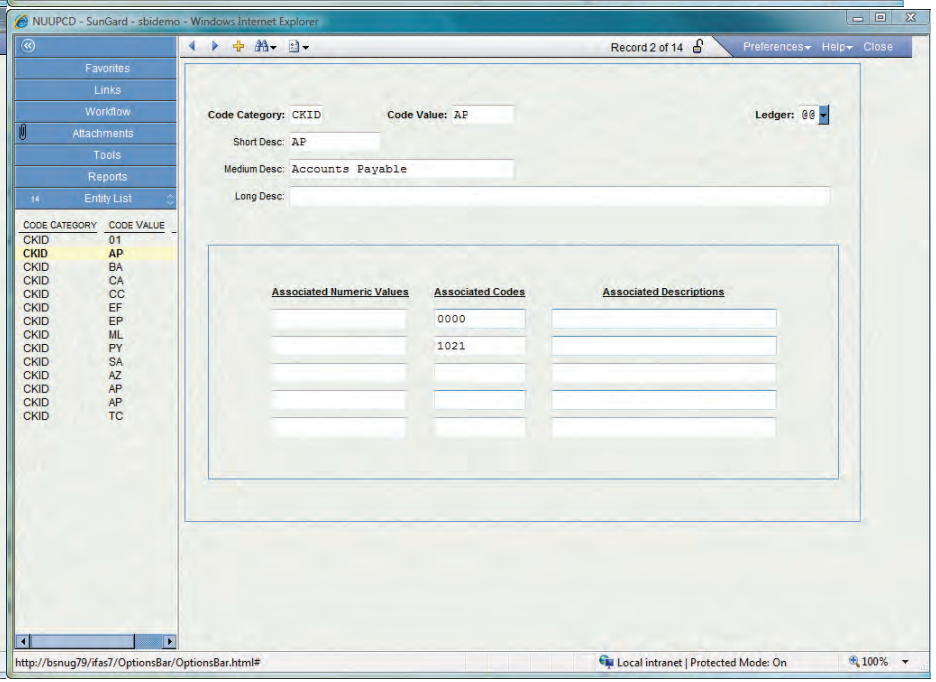
The only fields which must be filled in are Code Category and Code Value. Associated Code #1 will contain the Org. Key, if desired; Associated Code #2 will contain the Object Code, if desired. The Code Value field contains the left justified Check Stock ID or Fund ID and the Code Category contains the codes 'CKID' or 'FDID'. For our examples below, we will assume that the following entries have been made on the Common Codes form. PLEASE NOTE, IF THESE ENTRIES ARE NOT MADE OR ARE INCORRECT, INVALID POSTING MAY OCCUR TO THE GENERAL LEDGER.

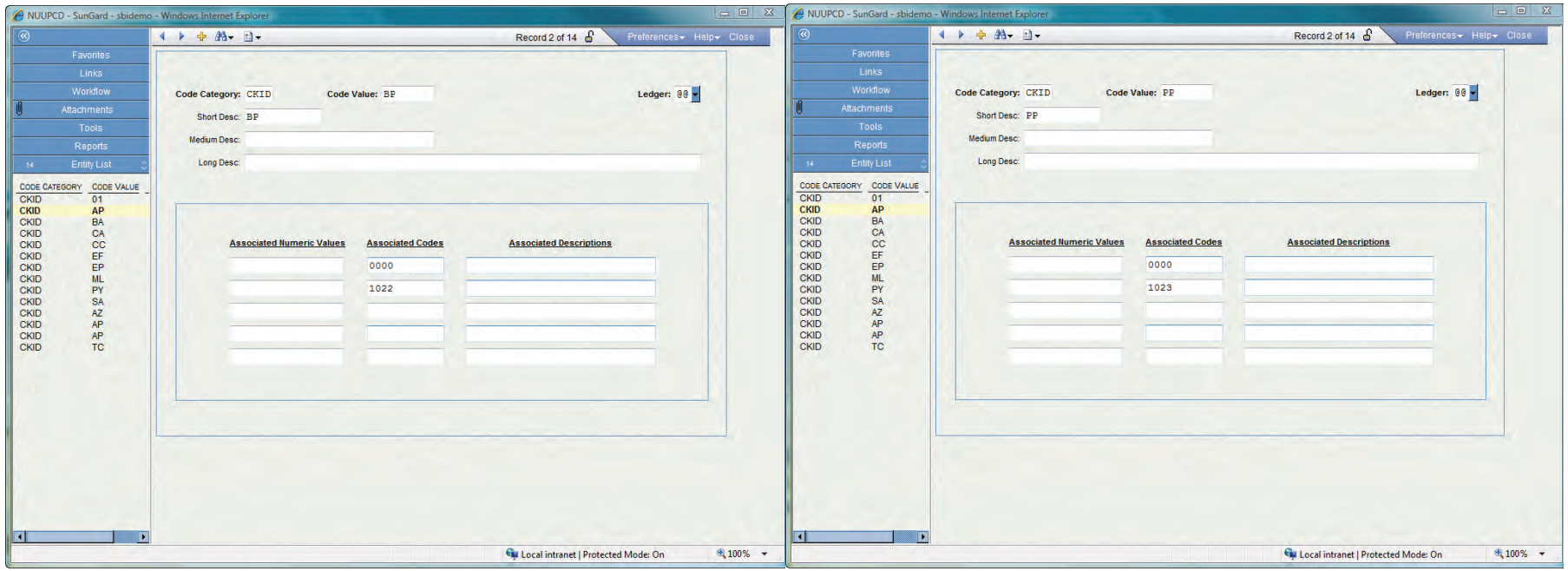
Code Category	Code Value	Associated Code #1	Associated Code #2
CKID	AP	0000	1021
CKID	BP	0000	1022
CKID	PP	0000	1023
FDID	10	0010	
FDID	20	0020	
FDID	30	0030	


NUUPCD - SunGard - sbidemo - Windows Internet Explorer
Record 3 of 93
Preferences Help Close
Favorites
Links
Workflow
Attachments
Tools
Reports
Entity List
Code Category: FDID
Code Value: 20
Ledger: @@
Short Desc:
Medium Desc: FUND ID 20
Long Desc:
Associated Numeric Values
Associated Codes
Associated Descriptions
0020
Done
Local intranet | Protected Mode: On
100%


NUUPCD - SunGard - sbidemo - Windows Internet Explorer
Record 9 of 93
Preferences Help Close
Favorites
Links
Workflow
Attachments
Tools
Reports
Entity List
Code Category: FDID
Code Value: 30
Ledger: @@
Short Desc:
Medium Desc: FUND ID 30
Long Desc:
Associated Numeric Values
Associated Codes
Associated Descriptions
0030
Local intranet | Protected Mode: On
100%


NUUPCD - SunGard - sbidemo - Windows Internet Explorer
Record 4 of 6
Preferences Help Close
Favorites
Links
Workflow
Attachments
Tools
Reports
Entity List
Code Category: FDID
Code Value: 10
Ledger: @@
Short Desc: Fund 10
Medium Desc: Fund 10 Administration
Long Desc: Fund 10 Administration Organization Key
Associated Numeric Values
Associated Codes
Associated Descriptions
0010
Fund 10 Admin Org Key
Local intranet | Protected Mode: On
100%


NUUPCD - SunGard - sbidemo - Windows Internet Explorer
Record 2 of 14
Preferences Help Close
Favorites
Links
Workflow
Attachments
Tools
Reports
Entity List
Code Category: CKID
Code Value: AP
Ledger: @@
Short Desc: AP
Medium Desc: Accounts Payable
Long Desc:
Associated Numeric Values
Associated Codes
Associated Descriptions
0000
1021
http://bsnug79/ifas7/OptionsBar/OptionsBar.html#
Local intranet | Protected Mode: On
100%

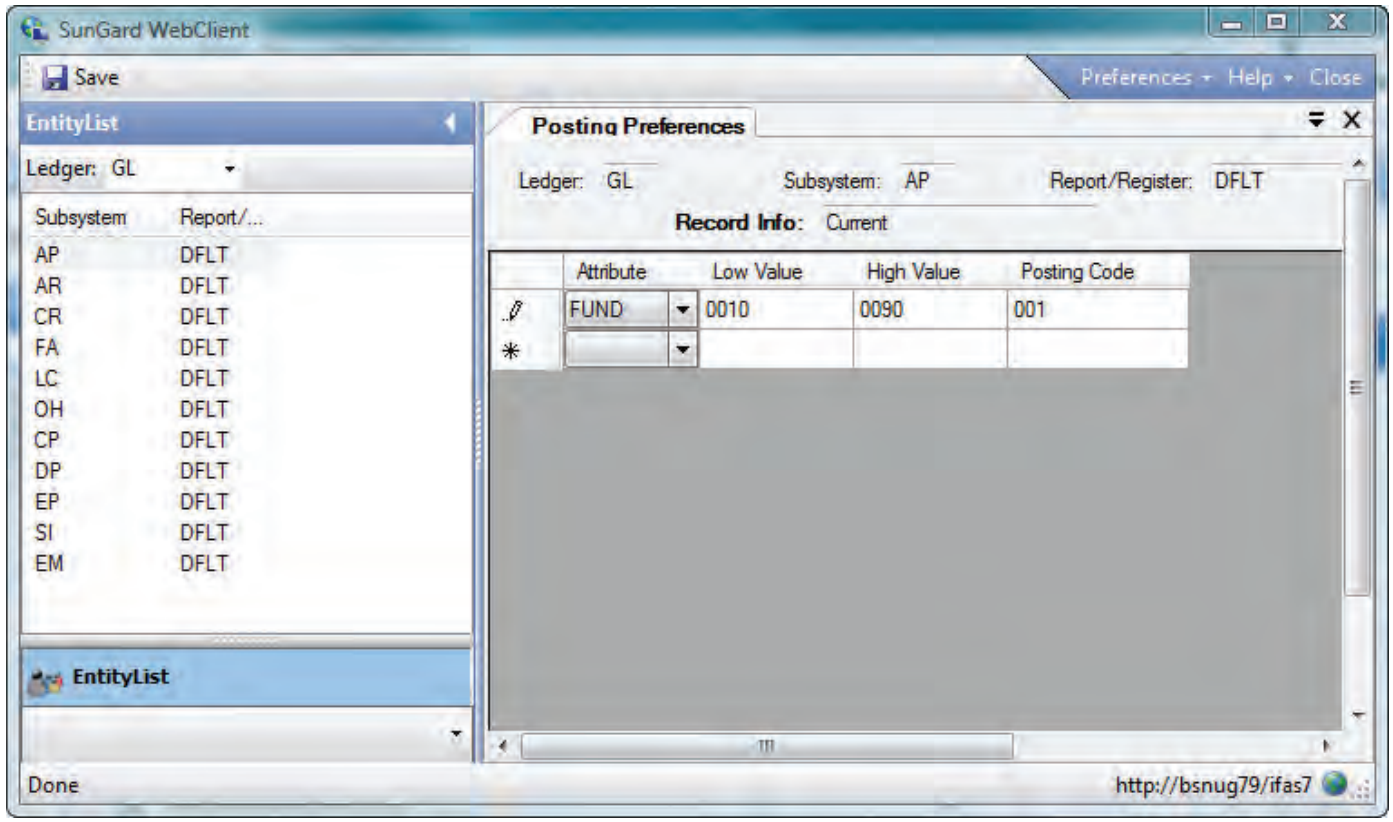


Sample Object Codes, Org. Keys, and Common Codes' entries have been defined. We are now ready to develop some examples. Each example will begin with a narrative of what the Accountant's perspective is. Next, some sample transactions are displayed. Then, how the Accountant would like those transactions to be posted. Finally, what the GL SUBSYSTEM INTERFACE form would contain to effect these desired transactions.

Posting Codes are used to direct the IFAS posting logic to create different automatic entries. Each code is associated with an Org. Key and Object Code to tell the system which fully qualified account to use in posting transactions to the GL. This is important when the client has defined multiple accounts in the GL which should receive postings from the various subsystems (i.e., Accounts Payable, Accounts Receivable, Payroll, etc.). Once different posting strategies have been defined, they are identified with various portions of the GL Structure. This allows the system, at data entry time, to automatically derive the correct posting code and place it in the set file. Subsection C of this section, Posting Preferences, describes the identification process. When entries are distributed to the GL, the detail transactions will be summarized at the LEVEL (L) defined for the posting code used and posted to the defined Org. Key and Object Code. In the examples that follow, effective use of posting codes is discussed, varying the situations.

6.2.1 Example A: ACCOUNTS PAYABLE

Narrative: The Accountant would like each Org. Key to be self balancing, with the system generating automatic entries to Object Code 1025 (Intra-fund Clearing Account) so that the department to fund relationship is represented. We will assume, as is typically the case, that there is only one Posting Code on A/P transactions, '01'. Further, assume that Discounts Earned are to be represented when the checks are actually printed. Further, assume that Discounts Earned are held in the Fund level Org. Key. Finally, we want Object Code 1029 to represent the relationship of funds to the bank accounts.



Sample Transactions

Org. Key	Object Code	Gross $ Amt	Disc. $ Amt	Net Amt	Check Stock	Posting Code
=======	=====	=====	=====	=====	======	
1000	8801	10	1	9	AP	01
1000	8802	20	2	18	AP	01
3000	8801	40	4	36	AP	01

What the Accountant Wants:

POSTED WHEN THE EXPENSES ARE DISTRIBUTED

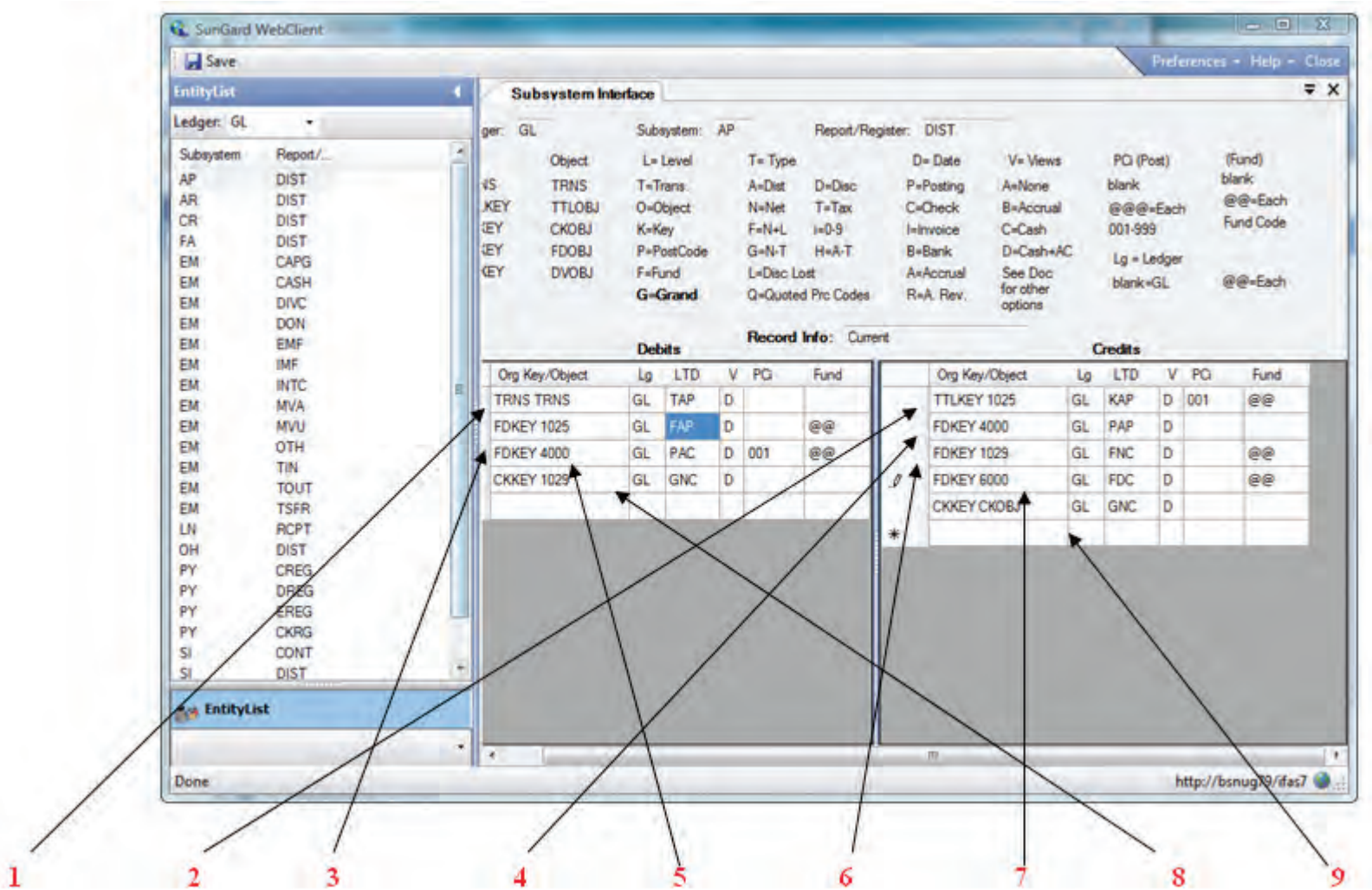
Dept One – Gen Expense One (1000-8801)	10	**1**		
Dept One – Gen Expense Two (1000-8801)	20	**1**		
Dept One – Intra Clearing (1000-1025)			30	**2**
Gen Fund 10 – Intra Clearing (0010-1025)	30	**3**		
Gen Fund 10 – AP (0010 – 4000)			30	**4**
Dept Three – Gen Expense One (3000-8801)	40	**1**		
Dept Three – Intra Clearing (3000-1025)			40	**2**
Rest Fund 20 – Intra Clearing (0020 – 1025)	40	**3**		
Rest Fund 20 – AP (0020 – 4000)			40	**4**

POSTED ONLY WHEN CHECKS ARE WRITTEN

Gen Fund 10 – AP (0010-4000)	30	**5**		
Gen Fund 10 – Disc Earned (0010-6000)			3	**6**
Gen Fund 10 – COC (0010 – 1029)			27	**7**
Rest Fund 20 – AP (0020-4000)	40	**5**		

	Rest Fund 20 – Disc Earned (0020-6000)			4	**6**
	Rest Fund 20 – COC (0020-1029)			36	**7**
Cash Admin – COC (0000-1029)		63	**8**		
	Cash Admin – Cash (0000-1021)			63	**9**

The GL SUBSYSTEM INTERFACE Form:



In the above screen, we are telling the posting logic to post the debit expense entry for the individual transactions; post a corresponding credit to the 'Intra-fund Clearing Account' (also, sometimes called 'intra-fund clearing' or 'clearing account'); and post fund total entries to recognize the Accounts Payable obligation and corresponding debit to the 'Intra-fund Clearing Account'. Finally, when checks are written, we recognize the discount and decrease the cash in the Cash Administration Org. Key with corresponding clearings to the Accounts Payable obligations in the separate Fund Org. Keys.

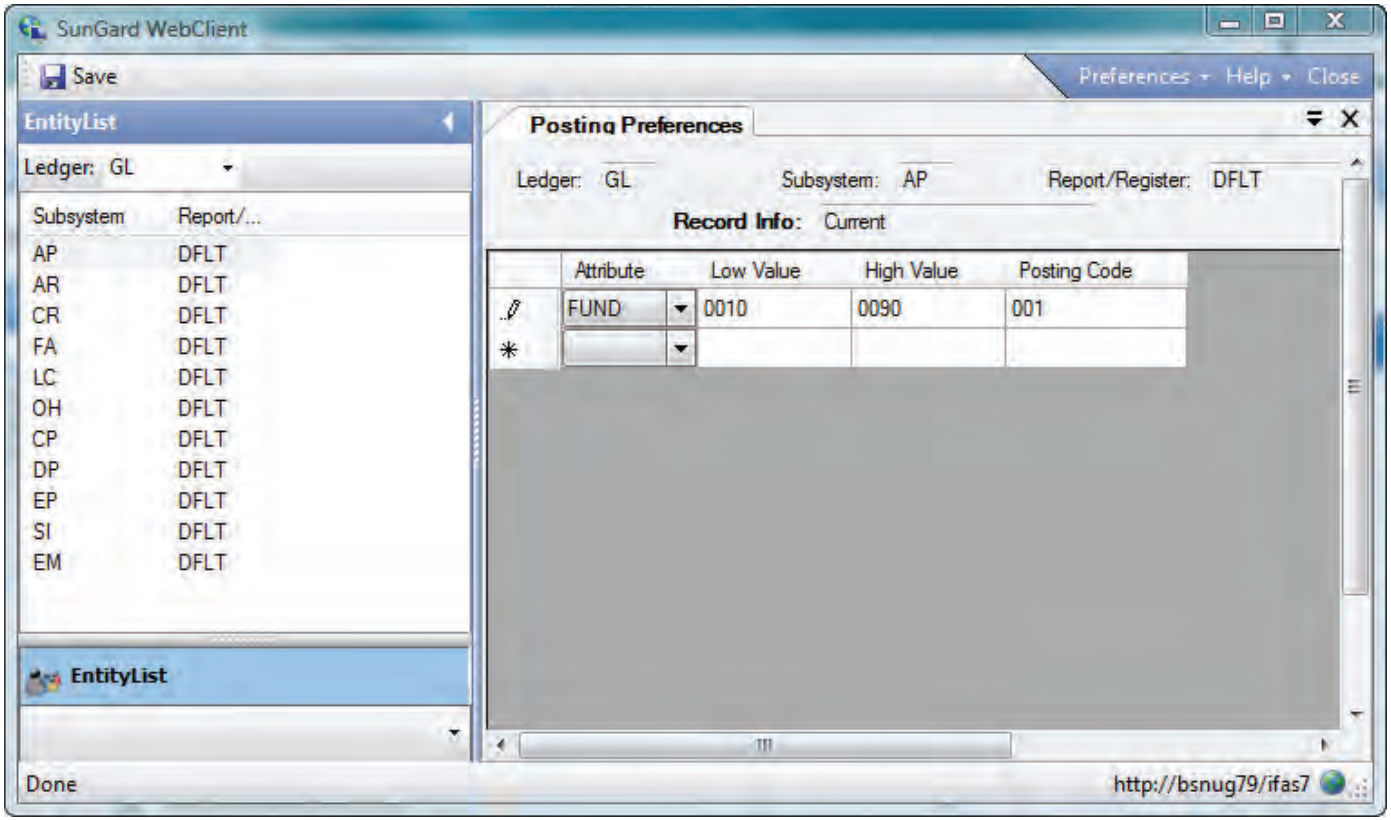
Note that with only one posting code, we can basically ignore the entry of a Posting Code in our records. If we had more than one Accounts Payable Object Code to which we wanted to post entries, we would likely set up separate Posting Codes to direct entries to the different A/P accounts. We will see such an example when Accounts Receivable is discussed, where multiple Accounts Receivable clearing accounts are used.

6.2.2 Example B: ACCOUNTS PAYABLE

Narrative: The Accountant would like each Fund to be self balancing. As in Example 1, we will assume there is only one Posting Code and Object Code 1029 will be used to show the inter-fund transfer between the separate Fund Org. Keys and the Cash Org. Key.

Sample Transactions

Org. Key	Object Code	Gross $ Amt	Disc. $ Amt	Net Amt	Check Stock	Posting Code
======	=====	=====	=====	=====	=====	======
1000	8801	10	1	9	AP	01
1000	8802	20	2	18	AP	01
3000	8801	40	4	36	AP	01



What the Accountant Wants:

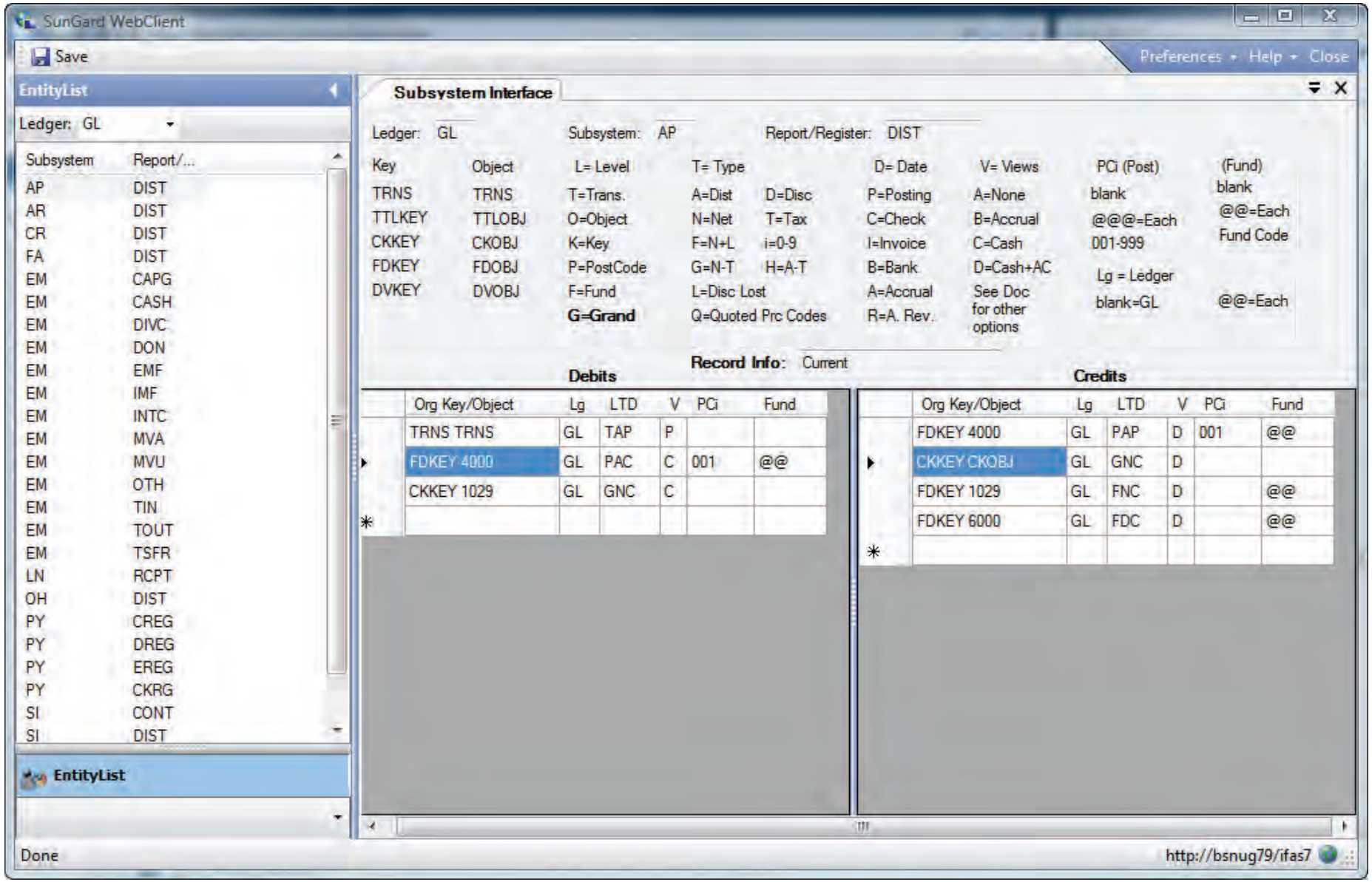
POSTED WHEN THE EXPENSES ARE DISTRIBUTED

Dept One – Gen Expense One (1000-8801)	10	
Dept One – Gen Expense Two (1000-8802)	20	
Gen Fund 10 – AP Gen (0010-4000)		30
Dept Three – Gen Expense One (3000-8801)	40	
Rest Fund 20 – AP Gen (0200-4000)		40

POSTED ONLY WHEN CHECKS ARE WRITTEN:

Gen Fund 10 – AP Gen (0100-4000)	30	
Gen Fund 10 – Disc Earned (0100-6000)		3
Gen Fund 10 – COC (0010-1029)		27
Rest Fund 20 – AP Gen (0020-4000)	40	
Rest Fund 20 – Disc Earned (0020-6000)		4
Rest Fund 20 – COC (0020-1029)		36
Cash Admin – COC (0000-1029)	63	
Cash Admin – Cash (0000-1021)		63

The GL SUBSYSTEM INTERFACE Form:



In the above screen, we are telling the posting logic to post the debit expense entry for the individual transactions and post a corresponding credit to Accounts Payable at the Fund level Account. Finally, when checks are written, we recognize the discount and decrease the cash in the Cash Administration Org. Key with corresponding clearings to the Accounts Payable obligations in the separate Fund Org. Keys.

Note that with only one posting code, we can basically ignore the Posting Code in our entries. If we had more than one Accounts Payable Object Code to which we wanted to post entries, we would likely set up separate Posting Codes to direct entries to the different A/P accounts. We will see such an example when we discuss Accounts Receivable, where multiple Accounts Receivable clearing accounts are used.

6.2.3 Example C: SPECIAL CASH/ACCRUAL ACCOUNTS PAYABLE

Narrative: In those circumstances where both Accrual and Cash Balances are to be posted, the system must distinguish between Immediate Pay and Void, Typed, Reversed distributions representing cash postings to the General Ledger and Open Hold distributions representing accrual postings until the actual check disbursement is made. (See the Accounts Payable User Guide for additional information on these three types of distributions.) The postings from Immediate Pay and Void, Typed, Reversed are defined with the 'AP/DIST' entries as described in examples 1 and 2. It is the user's choice as to whether postings are made to the General Ledger A/P Control Account and if Cash would be posted on the Check Date or the Posting Date. The Open Hold postings are defined with an 'OH/DIST' entry using both the Accrual and Accrual Reversal Date definitions. Each Fund is to be self balancing. As in Example 2, it will be assumed there is only one Posting Code and Object Code 1029 will be used to show the inter-fund transfer between the separate Fund Org. Keys and the Cash Org. Key.

Sample Transactions

Org. Key	Code	Object $ Amt	Gross $ Amt	Disc. $ Amt	Net Stock	Check Posting Code
1000	8801	10	1	9	AP	01
1000	8802	20	2	18	AP	01
3000	8801	40	4	36	AP	01

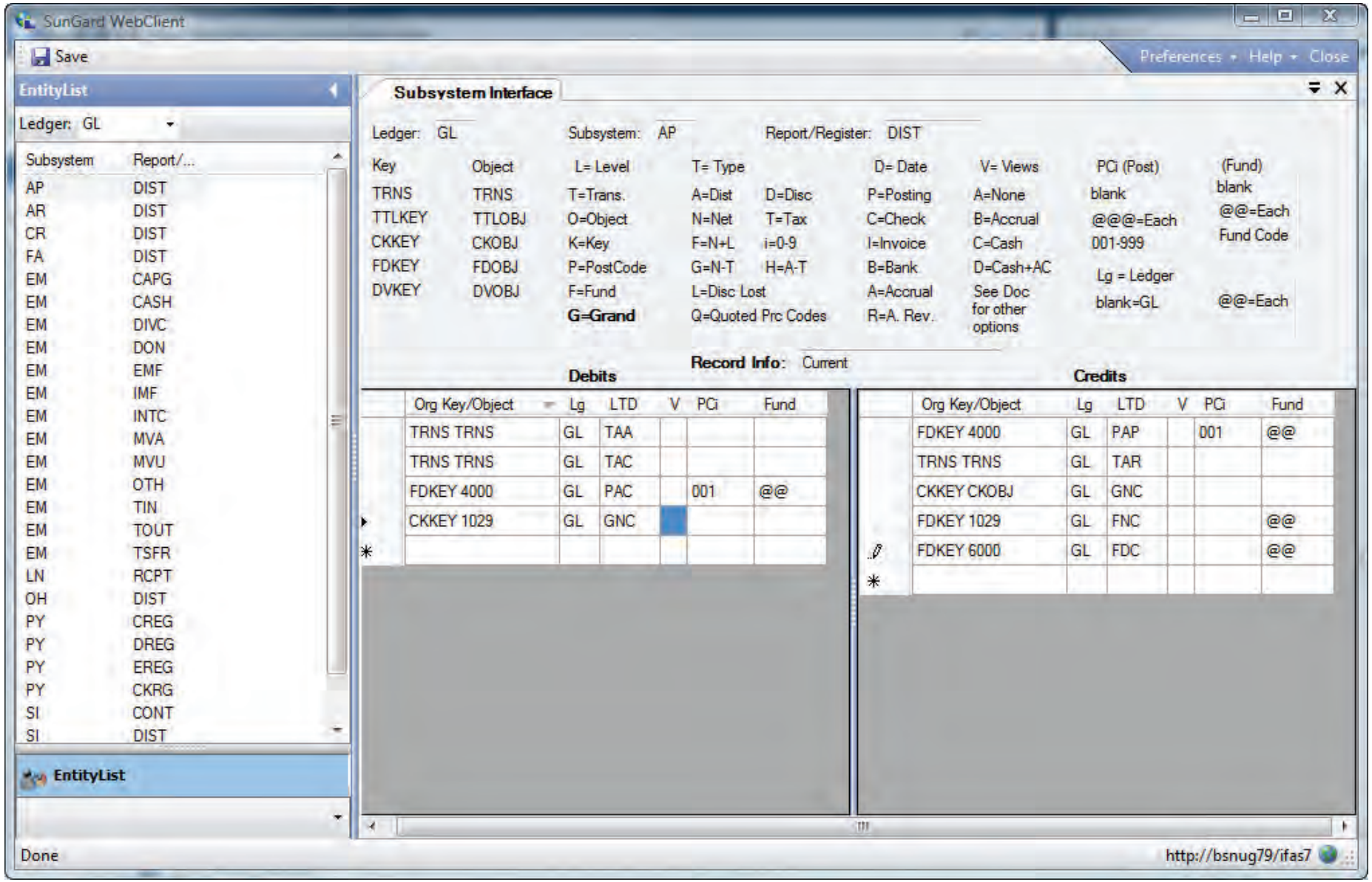
What the Accountant Wants:

POSTED WHEN THE EXPENSES ARE DISTRIBUTED

Dept One – Gen Expense One (1000-8801)	10	Accrual
Dept One – Gen Expense Two (1000-8802)	20	Accrual
Gen Fund 10 – AP (0010-4000)		30
Dept Three – Gen Expense One (3000-8801)	40	Accrual

Rest Fund 20 – AP (0020-4000)		40	
POSTED ONLY WHEN CHECKS ARE WRITTEN			
Dept One – Gen Expense One (1000-8801)	10	Cash	
Dept One – Gen Expense Two (1000-8802)	20	Cash	
Dept One – Gen Expense One (1000-8801)		10	Reverse Accrual
Dept One – Gen Expense Two (1000-8802)		20	Reverse Accrual
Gen Fund 10 - AP (0010-4000)	30		
Gen Fund 10 – Disc Earned (0010-6000)		3	
Gen Fund 10 – COC (0010-1029)		27	
Rest Fund 20 – AP (0020-4000)	40		
Rest Fund 20 – Disc Earned (0020-6000)		4	
Rest Fund 20 – COC (0020-1029)		36	
Cash Admin – COC (0000-1029)	63		
Cash Admin – Cash (0000-1021)		63	

The GL SUBSYSTEM INTERFACE Form:



6.2.4 Example D: RECEIVING AND ACCOUNTS PAYABLE

Narrative: At the time an item is received in Purchasing, the Accountant wishes to make postings to accrue expenses in the General Ledger, create a Receiving Liability, and relieve the Encumbered balances. Each Fund is to be self balancing. As in Example 2, it will be assumed there is only one Posting Code and Object Code 1029 will be used to show the inter-fund transfer between the separate Fund Org. Keys and the Cash Org. Key.

Sample Transactions

Org. Key	Object Code	Gross $ Amt	Disc. $ Amt	Net $ Amt	Check Stock	Posting Code
1000	8801	10	1	9	AP	01
1000	8802	20	2	18	AP	01
3000	8801	40	4	36	AP	01

What the Accountant Wants:

POSTED WHEN ITEMS ARE RECEIVED IN PURCHASING

Dept One – Gen Expense One (1000-8801)	10	Accrual
Dept One – Gen Expense Two (1000-8802)	20	Accrual
Gen Fund 10 – Receiving Clearing (0010-4010)	30	
Dept Three – Gen Expense One (3000-8801)	40	Accrual
Rest Fund 20 – Receiving Clearing (0020-4010)	40	

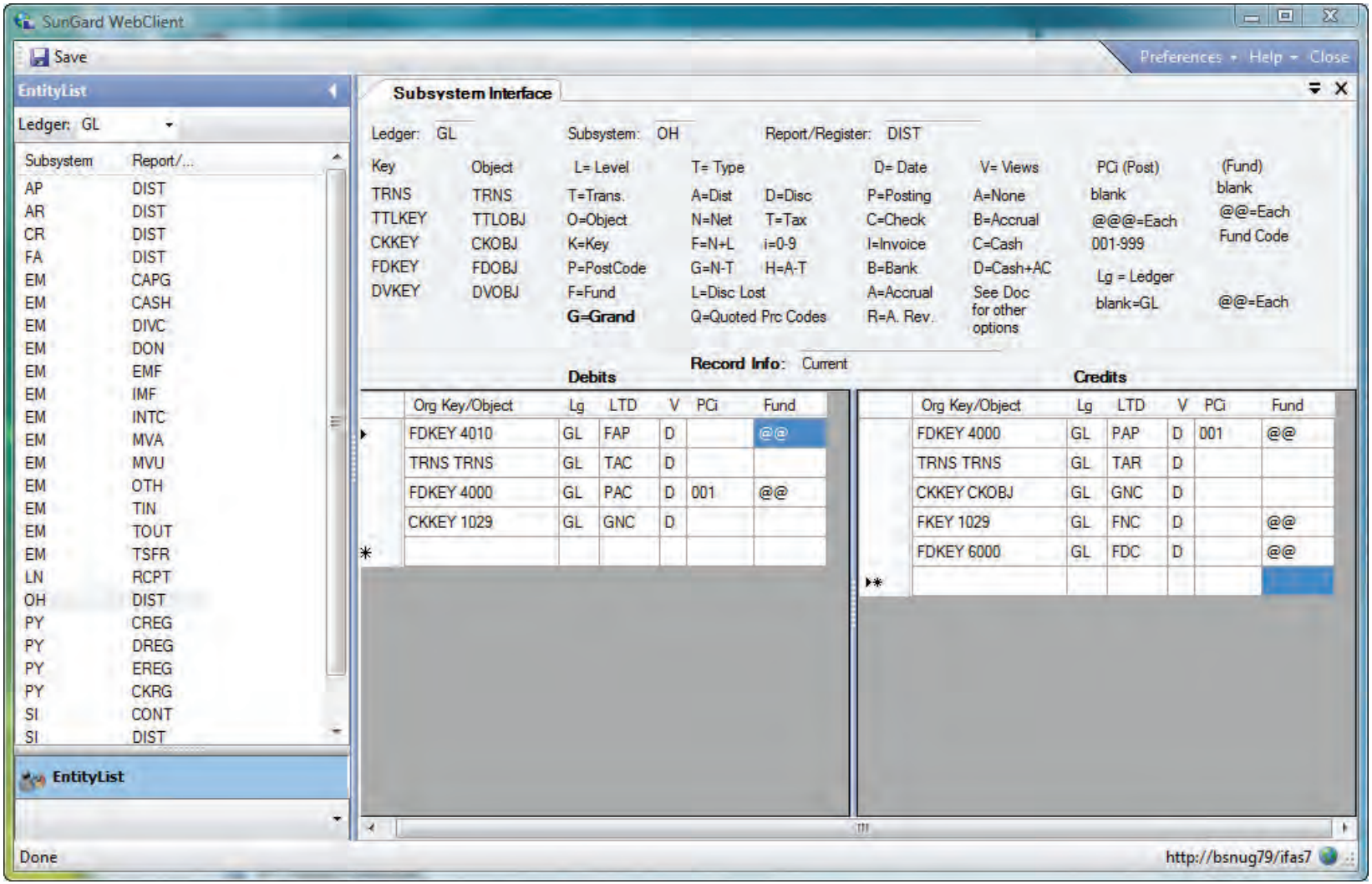
The GL SUBSYSTEM INTERFACE Form:

POSTED WHEN THE INVOICE IS RECEIVED AND EXPENSES ARE DISTRIBUTED

Gen Fund 10 – Receiving Clearing (0010-4010)	30	
Gen Fund 10 – AP (0010-4000)		30
Rest Fund 20 – Receiving Clearing (0020-4010)	40	

Rest Fund 20 – AP (0020-4000)		40	
POSTED ONLY WHEN CHECKS ARE WRITTEN			
Dept One – Expense One (1000-8801)	10	Cash	
Dept One – Expense Two (1000-8802)	20	Cash	
Dept One – Expense One (1000-8801)		10	Reverse Accrual
Dept One – Expense Two (1000-8802)		20	Reverse Accrual
Dept Three – Expense One (3000-8801)	40	Cash	
Dept Three – Expense One (3000-8801)		40	Reverse Accrual
Gen Fund 10 – AP (0010-4000)	30		
Gen Fund 10 – Disc Earned (0010-6000)		3	
Gen Fund 10 – COC (0010-1029)		27	
Rest Fund 20 – AP (0020-4000)	40		
Rest Fund 20 – Disc Earned (0020-6000)		4	
Rest Fund 20 – COC (0020-1029)		36	
Cash Admin – COC (0000-1029)	63		
Cash Admin – Cash (0000-1021)		63	

The GL SUBSYSTEM INTERFACE Form:



6.2.5 Example E: ACCOUNTS RECEIVABLE

Narrative: The Accountant would like each Department Org. Key to be self balancing with the system generating automatic entries to Object Code 1025 so that the department to fund relationship is represented. We will assume that transactions are coded with two Posting Code values, 01 and 02. When '01' is placed on a transaction, the Accounts Receivable clearing Object Code is to be 3001; when '02' is placed on a

transaction, the Accounts Receivable clearing Object Code is to be 3002. Since discounts will be recorded in the Accounts Payable examples, they will not be included here. Because individual transactions will be recorded in the Accounts Receivable database, only Object Code control totals will be recorded in the General Ledger.

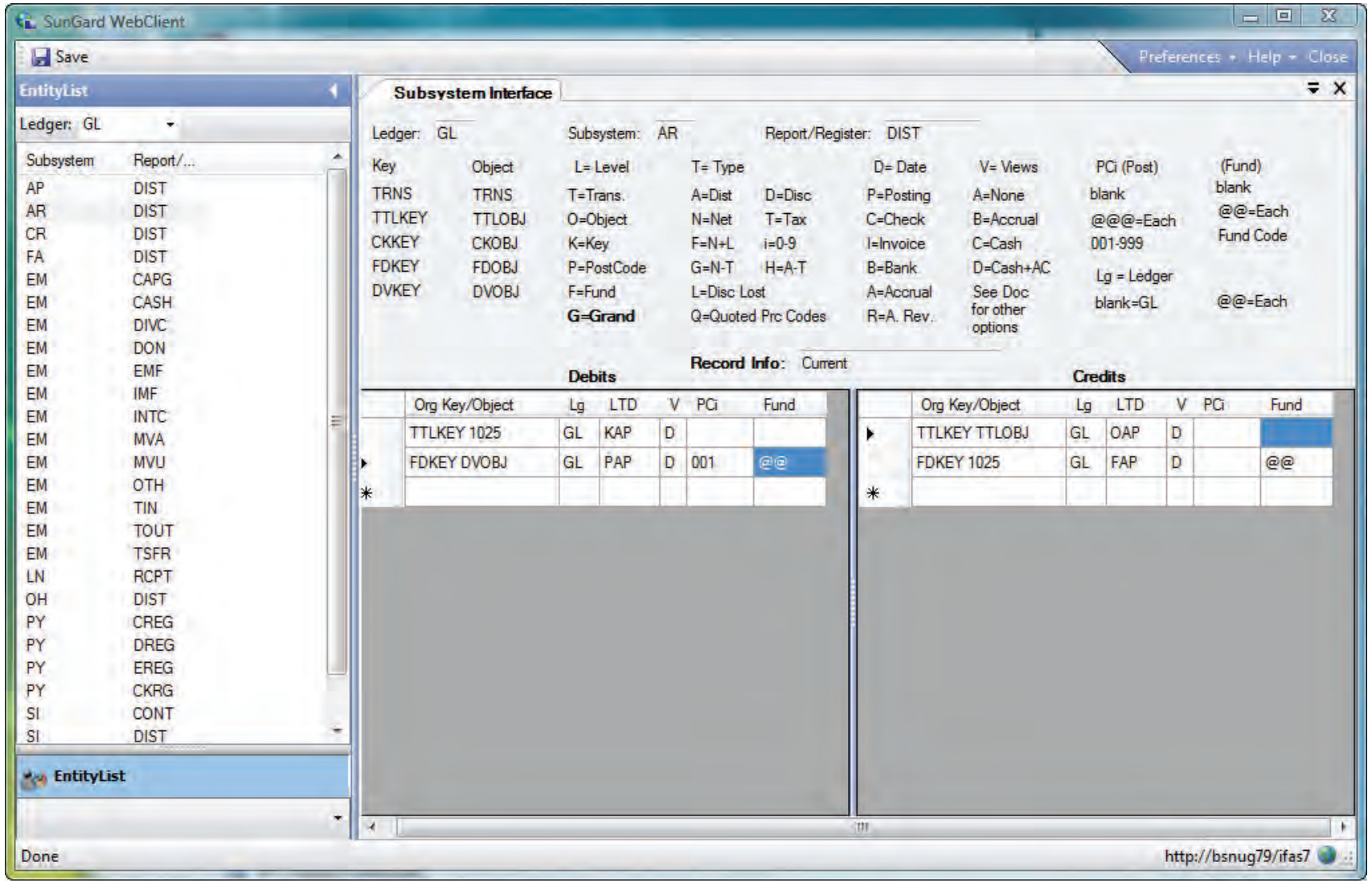
Sample Transactions:

Org. Key	Object Code	Gross $	Posting Amt Code
1000	6001	10	01
1000	6002	20	01
1000	6001	25	02
1000	6002	35	02
3000	6001	40	01

What the Accountant Wants:

Gen Fund 10 – AR 1 (0010-3001)	30	
Gen Fund 10 – AR 2 (0010-3002)	60	
Gen Fund 10 – Intra Clearing (0010-1025)		90
Rest Fund 20 – AR 1 (0020-3001)	40	
Rest Fund 20 – Intra Clearing (0020-1025)		40
Dept One – Intra Clearing (1000-1025)	90	
Dept One – Rev One (1000-6001)		35
Dept One – Rev Two (1000-6002)		55
Dept Three – Intra Clearing (3000-1025)	40	
Dept Three – Rev One (3000-6001)		40

The GL SUBSYSTEM INTERFACE Form:



In the above screen, we are telling the posting logic to post the credit (revenue entry) for the Object Code totals; post a corresponding debit to the 'Intra-fund Clearing Account'; post fund total entries for each Posting Code to the appropriate Accounts Receivable Object Code; and, post the other side (credit) to the 'Intra-fund Clearing Account' at the fund level.

6.2.6 Example F: ACCOUNTS RECEIVABLE

Narrative: The Accountant would like each Fund Org. Key to be self balancing. We will assume that transactions are coded with two Posting Code values, 01 and 02. When '01' is placed on a transaction, the Accounts Receivable clearing Object Code is to be 3001; when '02' is placed on a transaction, the Accounts Receivable clearing Object Code is to be 3002. Since discounts were used in the Accounts Payable examples, they will not be included here. Since individual transactions are posted to the Accounts Receivable database, only Object Code total entries will be posted to the General Ledger.

Sample Transactions:

Org. Key	Object Code	Gross $ Amt	Posting Code
1000	6001	10	01
1000	6002	20	01
1000	6001	25	02
1000	6002	35	02
3000	6001	40	01

What the Accountant Wants:

Gen Fund 10 – AR 1 (0010-3001)	30	
Gen Fund 10 – AR 2 (0010-3002)	60	
Rest Fund 20 – AR 1 (0020-3001)	40	
Dept 1 – Rev 1 (1000-6001)		35
Dept 1 – Rev 2 (1000-6002)		55
Dept 3 – Rev 1 (3000-6002)		40

The GL SUBSYSTEM INTERFACE Form:

In the above screen, we are telling the posting logic to post the credit (revenue entry) for the Object Code totals; and, post a corresponding debit to the appropriate Accounts Receivable clearing Object Code (based on Posting Code) at the fund level.

6.2.7 Example G: SPECIAL CASH/ACCRUAL ACCOUNTS RECEIVABLE:

Narrative: In those circumstances where both Accrual and Cash Balances are to be posted, the system must distinguish between revenues recognized when a receivable is established versus upon entry of a cash receipt. To accrue a revenue transaction the 'A' Date code is used. As in Example 2, we will assume there is only one Posting Code and we will use Object Code 1029 to show the inter-fund transfer between the separate Fund Org. Keys and the Cash Org. Key.

Sample Transactions

Org. Key	Object Code	Gross $ Amt	Posting Code
1000	6001	10	01
1000	6002	20	01
1000	6001	25	02
1000	6002	35	02
3000	6001	40	01

What the Accountant Wants:

Gen Fund 10 – AR One (0010 – 3001)	30	Accrual
Gen Fund 10 – AR Two (0010-3002)	60	Accrual
Rest Fund 20 – AR One (0020-3001)	40	Accrual
Dept One – Rev One (1000-6001)	35	Accrual
Dept One – Rev Two (1000-6002)	55	Accrual
Dept Three – Rev One (3000-6001)	40	Accrual

What the GL SUBSYSTEM INTERFACE Form would be:

6.2.8 Example H: CASH RECEIPTS

Narrative: The two types of Cash Receipts are those which are applied to an Accounts Receivable charge (i.e., coded to an Accounts Receivable Object Code on the transaction); and, those which do not relate to Accounts Receivable charges (i.e., coded to a revenue Object Code). To accommodate this, assume that the accountant makes use of Posting Code '01' to represent those Cash Receipts which are applied to Accounts Receivable and Posting Code '02' to represent those Cash Receipts which are coded to revenue accounts. Since all Cash Receipts coded to Accounts Receivable Object Codes are stored in the Accounts Receivable database, the accountant does not wish to store the detail level transactions in the General Ledger; thus, these Posting Code '01' transactions are going to be stored as control total entries. The second type of Cash Receipts, the ones coded to revenue Object Codes, are not kept in the Accounts Receivable database (typically); thus, the accountant will want to post detail level transactions to the General Ledger for each transaction having a Posting Code of '02'. This, in effect, creates a Cash Receipts Journal within the General Ledger.

For Posting Code '01' transactions, the accountant would like to post a credit total for each account, then a corresponding debit to Object Code 1029 to represent the funds' "claim on cash". At the Entity level, we would like to debit the Cash account and make a corresponding credit to Object Code 1029. When Posting Code '02' is used, we would like to post individual credit entries to the revenue object codes, a corresponding debit to the project clearing Object Code 1025, and a credit to the fund clearing Object Code 1025 to balance the fund. In addition, a debit entry should be made to the fund level Object Code 1029 to represent the funds' "claim on cash" in the Cash Administration Org. Key. At the Entity level, we would like to debit the Cash account and make a corresponding credit entry to Object Code 1029.

Sample Transactions

Org. Key	Object Code	Gross $ Amt	Posting Code
1000	6001	10	02
1000	6001	16	02
0010	3001	20	01
0010	3002	25	01
0010	3002	35	01
0020	3001	40	01

What the Accountant Wants:

Dept One – Intra Clearing (1000-1025) 26

Dept One – Rev One (1000-6001) 10

	Dept One – Rev One (1000-6001)	16
Gen Fund 10 – Intra Clearing (0010-1025)	80	
	Gen Fund 10 – AR One (0010-3001)	20
	Gen Fund 10 – AR Two (0010-3001)	60
	Gen Fund 10 – Intra Clearing (0010-1025)	106
Gen Fund 10 – Checking (0010-1029)	106	
	Rest Fund 20 – AR One (0020-3001)	40
Rest Fund 20 – Checking	40	
Cash Admin – Cash (0000-1021)	146	
	Cash Admin – Checking (0000-1029)	146

What the GL SUBSYSTEM INTERFACE Form would be

6.2.9 Example I: CASH RECEIPTS

Narrative: There are two types of Cash Receipts: Those which are applied to an Accounts Receivable charge (i.e., coded to an Accounts Receivable Object Code on the transaction); and, those which do not relate to Accounts Receivable charges (i.e., coded to a revenue Object Code). To accommodate this, we will assume that the accountant makes use of Posting Code '01' to represent those Cash Receipts which are applied to Accounts Receivable and Posting Code '02' to represent those Cash Receipts which are coded to revenue accounts. Since all Cash Receipts coded to Accounts Receivable Object Codes are stored in the Accounts Receivable database, the accountant does not wish to store the detail level transactions in the General Ledger; thus, these Posting Code '01' transactions are going to be stored as control total entries. The second type of Cash Receipts, the ones coded to revenue Object Codes, are not kept in the Accounts Receivable database (typically); thus, the accountant will want to post detail level transactions to the General Ledger for each transaction having a Posting Code of '02'. This, in effect, creates a Cash Receipts Journal within the General Ledger.

For Posting code '01' transactions, the accountant would like to post a credit total for each account, then a corresponding debit to Object Code 1029 to represent the funds' "claim on cash". At the Entity level, we would like to debit the Cash account and make a corresponding credit to Object Code 1029. When Posting Code '02' is used, we would like to post individual credit entries to the revenue object codes and debit entry to the fund level Object code 1029 to represent the funds' "claim on cash" in the Cash Administration Org. Key. At the Entity level, we would like to debit the Cash account and make a corresponding credit entry to Object Code 1029.

Sample Transactions

Org. Key	Object Code	Gross $ Amt	Posting Code
1000	6001	10	02
1000	6001	16	02
0010	3001	20	01
0010	3002	25	01
0010	3002	35	01
0020	3001	40	01

What the Accountant Wants:

Gen Fund 10 – Checking (0010-1029)	106	
Dept One – Rev One (1000-6601)		10
Dept One – Rev One (1000-6601)		16
Gen Fund 10 – AR One (0010-3001)		20
Gen Fund 10 – AR Two (0010-3002)		60
Rest Fund 20 – Checking (0020-1029)	40	
Rest Fund 20 – AR One (0020-3001)		40
Cash Admin – Cash (0000-1021)	146	
Cash Admin – Checking (0000-1029)		146

What the GL SUBSYSTEM INTERFACE Form would be

In the above screen, for Posting Code '01', the posting logic is being told to post a control total of all CR transactions to each Project, Object Code combination, a to the Cash account, and corresponding entries to Object Code 1029. For Posting Code '02' individual transactions are posted to each revenue account along with entries to Cash, and Object Code 1029.

6.2.10 Example J: SPECIAL CASH/ACCRUAL CASH RECEIPTS

Narrative: In those circumstances where both Accrual and Cash Balances are to be posted, the system must distinguish between revenues recognized when a receivable is established versus upon entry of a cash receipt (See Example 3 for Accounts Receivable). To recognize the Cash entry upon Cash Receipting, the 'R' and 'P' Date Codes are used to reverse the original revenue accrual and post the cash entry. As in Example 2, we will assume there is only one Posting Code and we will use Object Code 1029 to show the inter-fund transfer between the separate Fund Org. Keys and the Cash Org. Key.

Sample Transactions

Org. Key	Object Code	Gross $ Amt	Posting Code
1000	6001	10	02
1000	6001	16	02
0010	3001	20	01
0010	3002	25	01
0010	3002	35	01
0020	3001	40	01

What the Accountant Wants:

Dept One – Rev One (1000-6001)	10
Dept One – Rev One (1000-6001)	16
Gen Fund 10 – AR One (0010-3001)	20
Gen Fund 10 – AR Two (0010-3002)	60

Gen Fund 10 – Checking (0010-1029)	106		
Rest Fund 20 – AR One (0020-3001)		40	
Rest Fund 20 – Checking (0020-1029)	40		
Dept One – Rev One (1000-6001)		35	Cash
Dept One – Rev Two (1000-6002)		55	Cash
Dept One – Rev One (1000-6001)	35	Accrual Reversal	
Dept One – Rev Two (1000-6002)	55	Accrual Reversal	
Dept Three – Rev One (3000-6001)		40	Cash
Dept Three – Rev One (3000-6001)	40	Accrual Reversal	
Cash Admin – Cash (0000-1021)	146		
Cash Admin – Checking (0000-1029)		146	

What the GL SUBSYSTEM INTERFACE Form would be:

6.2.11 Example K: FIXED ASSET DEPRECIATION

Narrative: The accountant would like to post depreciation to the depreciation expense Object Code for the Department Org. Key. Since detail depreciation amounts are held in the Fixed Assets database, only control total amounts for each Department Org. Key are posted. A corresponding (by fund) is posted to the depreciation liability at the Fund Org. Key level. In this example, we also show the 'Intra-fund clearing' postings to 1025.

Sample Transactions

Org. Key	Object Code	Depreciation $ Amount
1000	8900	10
1000	8900	20
2000	8900	25

3000	8900	40

What the Accountant Wants

Dept One – Dep Exp (1000-8900)	30	
Dept One – Intra Clearing (1000-1025)		30
Dept Two – Dep Exp (2000-8900)	25	
Dept Two – Intra Clearing (2000-1025)		25
Gen Fund 10 – Intra Clearing (0010-1025)	55	
Gen Fund 10 – Dep (0010-4900)		55
Dept Three – Dep Exp (3000-8900)	40	
Dept Three – Intra Clearing (3000-1025)		40
Rest Fund 20 – Intra Clearing (0020-1025)	40	
Rest Fund 20 – Dep (0020-4900)		40

In the above screen, we are telling the posting logic to post, at the Department Org. Key total level, the depreciation expense and make a corresponding, opposite signed entry to 'Intra-fund Clearing'. At the Fund Org. Key level, we recognize the depreciation liability and make the corresponding 'Intra-fund Clearing' entry.

6.2.12 Example L: FIXED ASSET DEPRECIATION

Narrative: The accountant would like to post depreciation to the depreciation expense Object Code for the Department Org. Key. Since detail depreciation amounts are held in the Fixed Assets Database only control total amounts for each Department Org. Key are posted. A corresponding (by fund) is posted to the accumulated depreciation at the Fund Org. Key level.

Sample Transactions

Org. Key	Object Code	Depreciation $ Amount
1000	8900	10

1000	8900	20
2000	8900	25
3000	8900	40

What the Accountant Wants:

Dept One – Dep Exp (1000-8900)	30	
Dept Two – Dep Exp (2000-8900)	25	
Gen Fund 10 – Dep (0010-4900)		55
Dept Three – Dep Exp (3000-8900)	40	
Rest Fund 20 – Dep (0020-4900)		40

In the above screen, the posting logic is directed to post the depreciation expense, at the Department Org. Key total level and make a corresponding, opposite signed, entry to the accumulated depreciation at the Fund Org. Key level.

GL Setup

Posting Preferences (GLUTSPPP) GL334

Posting preferences for Order Filling and SI Set Receiving are listed under the **SI** subsystem.

Posting preferences for Inventory Exception Posting is listed under the **SC** subsystem.

Posting Codes may be based off of typical Key, Obj, Key parts, and Obj parts as well as WHSE and CUST (transaction code SITR:xx).

Posting Strategies (GLUTSPSI) GL333

GL / SI / DIST Order Filling

GL / SI / RCVGSI Set Receiving

GL / SI / CONT Inventory Exception Posting (assumes shortage - reducing inventory)

Distribution = “Base Price” (i.e., FIFO, LIFO, AVG, RP)

Charge = additional charges (i.e., WO, CC, constant)

Discount = total discounts (customer & product - product applied first, then customer)

Tax = tax amount (applied last, after all charges & discounts)

Example:

Order Qty = 10

FIFO cost = 1.00

WO = 5%

Discount = 10%

Tax = 6%

	“FIFO * 1.03”	“FIFO + WO”
Distribution	10.00	10.00
Charge	0.30	0.50
Discount	1.03	1.05
Tax	0.56	0.57
TOTAL	9.83	10.02

Posting Type	**Value**
A	Distribution + Tax + Charge
N	Distribution + Charge - Discount + Tax
C	Charge
D	Discount
T	Tax

To post the “Base Price” (distribution amount) alone, create a user defined calc (GLUTSPSC) for the SI subsystem that defines Type Code “B” as Calculation “A”. You can then use “B” as the Posting Type in GLUTSPSI.

PAYROLL Postings

The Payroll system requires many additional definition capabilities due to the complexity and broad range of postings which are desired by the various clients. All postings are divided into the following three distribution registers:

EREG Earnings - Regular, Overtime, Vacation, Sick, Compensatory Time Off and optionally other Employer provided Direct Contributions.

DREG Deductions - Individual employee payroll deductions.

CREG Contributions - Employer provided fringe benefits.

In its simplest form, the Earnings Register defines postings to DR gross wage/salary expense account(s) and CR net pay to the cash account while the Deduction Register (DREG) identifies the CR posting to a payroll liability account(s). The Contribution Register (CREG) defines postings to DR Fringe Benefit expense account(s) and CR Fringe Benefit expense to payroll liability account(s). There are many different methodologies which may be defined to provide postings to the General Ledger at a particular Level (L), define inter/intrafund balancing, and Account posting requirements. The discussion below will provide details on the use of each option available.

General Rules for ALL Payroll Registers:

Ledger (Lg):

If this entry should occur for only a specific GL Ledger, enter the appropriate two-character Ledger Code in this field. A code of '@@' will allow the GLUTSPSI entry to occur for all ledgers.

Date (D):

This field determines the posting date of the General Ledger Transaction. Codes 'P' and 'C' are valid. These dates are taken from the Payroll Period Definition Screen, PYUPPR.

Posting Code (PCi):

Generally this field is left 'blank' or has '@@@' defined. Under these circumstances, the entry will occur for all Contribution, Deduction, and Hour (CDH) definitions. However, in certain situations, individual CDH's must post to unique accounts rather than the standard posting. For these special postings, enter the last three (3) digits of the CDH definition. The system will then IGNORE any other GLUTSPSI entry on the same 'side' with a 'TTLOBJ' or 'TRNS' entry where the PCi is 'blank' or '@@@'.

For example, the following GLUTSPSI entry:

	Org Key / Object	**LgLT**	D	PCi	**Fund**
#1	FDKEY FDOBJ	@@	F7	P	@@

#2	TTLKEY TTLOBJ	@@	K7	P	@@@
#3	TTLKEY 1038	@@	K7	P	567

would always post line #1. Line #2 occurs for each CDH EXCEPT 567. Line #3 occurs only for CDH 567.

Fund (Fund):

As in the other subsystems, if the GLUTSPSI entry is 'FDKEY' or 'FDOBJ' the Fund field must have '@@' or a specific General Ledger Fund Code. Likewise if the entry is to occur for only a specific Fund then enter the appropriate Fund Code.

Other Considerations:

When posting any Payroll Register, if the GL Object field, DR or CR, within a CDH definition has 'NOPOST', the record is IGNORED.

Units or Hours postings are made for only 'TTLKEY TTLOBJ' or 'TRNS TRNS' postings.

If the JL Object is not specified within the CDH definition, no JL Posting occurs. This is true though a JL Object is defined on the Pay String of the Pay Assignment Definition.

Each GLUTSPSI entry, when posted, is net of all qualified activity. For example, if John Smith had earnings for 80 hours @$5.00 per hour and -2 hours @$5.00 per hour, Payroll would create two records, one for $400 and the other for -$10. When the expense is posted to the General Ledger, the transaction will be the net of $390.

Earnings Register (EREG) - GL/PY/EREG

EREG provides the greatest number of options for identifying the different levels and types of gross and net pay postings to the General Ledger. Not all options will be used within one payroll posting definition. In addition, the Level (L) and Type (T) defined will determine the information placed in the General Ledger transaction description field.

Total for each:Level (L)Transaction Description

Individual EmployeeTEmployee Name

Each different ObjectOObject Description

Each different Org. KeyKOrg. Key Title

Each different CDHPCDH Description

Fund AdministrationFFund Description

G “Register Name”

Type of Posting:Type (T)Transaction Description

Regular Salary/Wages1REG

Overtime Pay2O/T

Vacation Pay3VAC

Holiday Pay4HOL

Sick Pay5SICK

Compensatory Time Off Pay6CTO

Gross Pay7GRS

Net Pay8NET

Gross Pay + Contributions9TEX

Gross Pay + Direct Contributions0TOT

For example, the following GLUTSPSI entry:

Org Key / Object	**LgLT**	**D**	**PCi**	**Fund**
TTLKEY TTLOBJ	@@	K7	P	

would post one total to each Org. Key, Object combination and have a General Ledger transaction description of:

"Org. Key Title GRS"

The Org. Key used during the posting process for any 'TTLKEY' entry will be pulled from the Employee Pay Assignment or from the Timecard record if an override is allowed.

The Object of Expense used during the posting process for any 'TTLOBJ' entry will be pulled from the Hours Definition or the Pay Assignment Pay String.

Deduction Register (DREG) - GL/PY/DREG

and

Contribution Register (CREG - GL/PY/CREG

The definition for both DREG and CREG are identical. The only difference is that DREG posts information related to employee deductions, while CREG posts employer provided Fringe Benefits or Contributions. They both provide only one Type (T) of posting. However, just as in EREG, the Level (L) and Type (T) defined will determine the information placed in the General Ledger transaction description field.

Total for each:

Level (L)Transaction Description

Individual EmployeeTEmployee Name

Each different ObjectOObject Description

Each different Org. KeyKOrg. Key Title

Each different CDHPCDH Description

Fund AdministrationFFund Description

G "Register Name"

Type of Posting:

Type (T) Transaction Description

Regular Salary/Wages1REG

The Org. Key used during the posting process for any 'TTLKEY' entry will be pulled from the Employee Pay Assignment or from the Timecard record if an override is allowed.

The Object of Expense used during the posting process for any 'TTLOBJ' entry will be pulled from the Deduction or Contribution Definition.

6.2.13 Example M: PAYROLL

Narrative: The accountant would like to post a total for Gross Salaries and Wages to each Department Org. Key. Employer Fringe Benefit Contributions are posted to two different Object Codes based on the Payroll Contribution number '001' = Type One and '002' = Type Two. At the Fund Administration level, Net Pay will be posted to Cash in Bank Payroll for each Fund. Employer Contributions will be separated from Employee Deductions into two different liability accounts and differentiated between the Contribution and Deduction numbers '001' and '002'. Each Org. Key will be self balancing with postings to the 1025 Object Code.

Sample Transactions

Org. Key	Object Code	Payroll $ Amount
1000	1801	1000
1000	8501	200
2000	8501	1500
2000	8502	300

0010	1023	1850
0010	4501	200
0010	4502	300
0010	4601	250
0010	4602	400

What the Accountant Wants:

Dept One – Wages (1000-8001)	1000		
Dept One – Type One (1000-8501)	200		
Dept One – Intra Clearing (1000-1025)		1200	
Dept Two – Wages (2000-8501)	1500		
Dept Two – Type Two (2000-8502)	300		
Dept Two – Intra Clearing (2000-1025)		1800	
Gen Fund 10 – Intrafund (0010-1025)	3000		
Gen Fund 10 – Type One (0010-4601)		250	Employee
Gen Fund 10 – Type Two (0010-4602)		400	Employee
Gen Fund 10 – Cash (0010-1023)		1850	
Gen Fund 10 – Type One (0010-4501)		200	Employer
Gen Fund 10 – Type Two (0010-4502)		300	Employer

One of the difficult portions in defining Payroll Postings is that each individual distribution screen may not represent a balanced set of entries. However, in total, all three distributions, EREG, CREG, and DREG, should be balanced. Note that in the above example, DREG posts only the credit to Employee Deduction Liability. EREG, posts a debit to Salaries and Wages for Gross Pay and a credit to Cash in Bank Payroll for the Net Pay.

6.2.14 Example N: PAYROLL

Narrative: The accountant would like to post a total for Gross Salaries and Wages to each Department Org. Key. Employer Fringe Benefit Contributions are posted to two different Object Codes based on the Payroll Contribution number '001' = Type One and '002' = Type Two. At the Fund Administration level, Net Pay will be posted to Cash in Bank Payroll for each Fund. Employer Contributions will be separated from Employee Deductions into two different liability accounts and differentiated between the Contribution and Deduction numbers '001' and '002'. Sample Transactions shown below.

Org. Key	Object Code	Payroll $ Amount
1000	1801	1000
1000	8501	200
2000	8501	1500
2000	8502	300
0010	1023	1850
0010	4501	200
0010	4502	300
0010	4601	250
0010	4602	400

What the Accountant Wants:

Dept One – Wages (1000-8001)	1000		
Dept One – Type One (1000-8501)	200		
Dept Two – Type One (2000-8001)	1500		
Dept Two – Type Two (2000-8502)	300		
Gen Fund 10 – Cash (0010-1023)		1850	
Gen Fund 10 – Type One (0010-4501)		200	Employer
Gen Fund 10 – Type Two (0010-4502)		300	Employer
Gen Fund 10 – Type One (0010-4601)		250	Employee
Gen Fund 10 – Type Two (0010-4602)		400	Employee

Payroll Posting Mapping - OPTIONAL

Under some circumstances, it may be necessary to direct unique, complex posting logic based on relationships within the GL Structure and/or Employee Pay Types. For instance, If an employee is full-time and the Hours Code is '123', then Earnings should be posted to a specific Object of Expense. Yet, when they are part-time with the same Hours Code, Earnings are posted to a different Object of Expense. In addition, this posting may effect GL and/or JL accounts.

IF THE SITE DOES NOT NEED THESE TYPES OF DEFINITIONS, POSTING MAPPING SHOULD NOT BE DEFINED.

This process uses a combination of the POSTING KEY/OBJ MAPPING form, shown below, and NUCLEUS Common Codes. The POSTING KEY/OBJ MAPPING form can be found by keying, **GLUTSPPM**.

Subsystem: Enter the two-character code for the appropriate subsystem:

EP - EarningsDP - Deductions

CP - Contributions.

Ledger: Enter the two-character code for the Ledger which will be affected by this Mapping Logic.

If: Specify the portion of the IFAS structure which should be used in analyzing the details associated with a transaction in the set to determine the Mapping Code to be used. Defined codes are:

KEYGL Org. KeyOBJGL Object Code

PAYHours CodeCNTContribution Number

DEDDeduction NumberM1Employee Type

XXXX Four-character code of either M2 Employee Unit

the GL Org. Part or Object Group M8 Employee Skill Code 3

is between: This field allows for the entry of a specific code which relates to the IFAS structure indicated in the 'If' field.

and: Enter the second code when defining a range, or leave blank if a singular item.

then the Mapping Code is: The specific Mapping Code which points to a NUCLEUS Common Code. The NUCLEUS Common Code defines the alternate GL and/or JL posting accounts to be used when the 'IF' criteria is met. (Common Code, Code Category 'GLxx' where 'xx' is the two-character subsystem and Code Value 'PMAPyyy' where 'yyy' is the Mapping Code) When there are multiple 'IF' conditions which need to be evaluated before a Mapping Code can be defined, then 'AND' is placed in this field to indicate that the next condition must also be met before finding a specific Mapping Code.

A maximum of 150 entries may be made on the Posting Mapping Screen for each Subsystem and Ledger.

Find Subsystem/Ledger: In BROWSE-mode, enter the desired two-character Ledger code and the two-character subsystem code to view the desired Mapping Code.

The following example shows that for Earnings, when an employee is Full-Time and the Hours Code is '123' then the user wants the JL Object Code to be '6215' rather than the default entry which is defined within the Payroll Hours Definition for Hours Code '123'. The Mapping Code desired is '001'. Note the Common Code entry below which defines the alternate JL Object Code.

The NUCLEUS Common Code for Posting Mapping has the following general definition:

Code Category: 'GLxx', where 'xx' is the two-character Subsystem Code on the Posting Mapping Form.

Code Value: 'PMAPyyy', where 'yyy' is the three-character Mapping Code on the Posting Mapping Form.

Associated Code #1: Alternate GL Org. Key to be used in posting.

Associated Code #2: Alternate GL Object Code to be used in posting.

Associated Code #3: Alternate JL Org. Key to be used in posting.

Associated Code #4: Alternate JL Object Code to be used in posting.

Each of the Associated Code entries are OPTIONAL. The user should enter only values if that portion of the structure is to have an alternate posting. If an Associated Code is left blank, then that portion of the GL/JL structure will be taken from the CDH definition, Pay Assignment, or Timecard entry depending upon the individual client use. Also, if 'NOPOST' is defined within the Payroll GL Object definition, Mapping is ignored.

The following example shows that for Mapping Code '001', an alternate JL Object Code will be used rather than whatever the standard JL Object Code is defined within Payroll.

6.3 Posting Preferences

Once Posting Codes have been defined within the GL SUBSYSTEM INTERFACE form, **GLUTSPSI**, Posting Preferences must be identified. However, Posting Codes are OPTIONAL for Payroll.

Note: A question has been added to masks GLUTSPPP and GLUTSPPM. "Would you like to load posting codes from a file." The file is in 378b CD-CODE-MSTR format. A common code tracks recovery files and errors. Category POST, code LOGLG334. The work log may be placed in the short description field. This will keep detailed logs for all DB transactions (the files are titled PL#####A, with A being a sequence letter and ##### a date code). If the word AUTO is placed in the medium description - auto recovery is turned on. This masks now require exclusive access to run. If you try to run when someone else is using it - you will get an error message along with the user id of the current user.

All posting codes are set up through the GL SUBSYSTEM INTERFACE screen (**GLUTSPSI**). The DEFAULT POSTING CODES screen (**GLUTSPPP**) is used in conjunction with **GLUTSPSI**. Posting Preferences eliminate the need to enter the Posting Code during data entry time by automatically determining the appropriate code for each transaction based on the data entered on the DEFAULT POSTING CODES screen.

Posting Preferences are set up by subsystem. When working in any of the subsystems, IFAS will pull the appropriate information from the DEFAULT POSTING CODES screen. Since individual data entry is not required from Fixed Assets it does not need this process.

When utilizing this feature, some factors are very important. The user must insure that all ranges of Org. Keys using the GL parts, etc., are defined. If not, some gaps may appear and the system will not be able to derive a posting code. In addition, every Posting Code defined on **GLUTSPSI** must also be defined on the DEFAULT POSTING CODES screen. If a Posting Code is not defined on the DEFAULT POSTING CODES screen, the Posting Code field on the data entry screen will remain blank after enter has been pressed. This will result in that transaction not being posted to the GL.

The order in which the posting codes are entered on the DEFAULT POSTING CODES screen determines the order the system will search to find the applicable posting code for a given transaction. Therefore, it is important to enter data on the DEFAULT POSTING CODES screen in a hierarchy from specific to general. In the case where there is only one posting code, only one entry is necessary on the DEFAULT POSTING CODES screen. For example, the user could define the Object Code as 01-9999; thus, encompassing all possible transactions.

Subsystem: A two-character code relating to the IFAS subsystem from which postings should be referenced. Appropriate codes are:

AP Accounts Payable

AR Accounts Reveivable

CR Cash Receipts paid against Accounts Receivable

LC Lonesome Cash Receipts (No A/R)

SI Stores Inventory

SC Stores Adjustments

EP Payroll Earnings

DP Payroll Deductions

CP Payroll Contributions

Ledger: Enter the two-character code for the Ledger which will be affected by this Posting Logic.

If: Specify the portion of the IFAS structure which should be used in analyzing the details associated with a transaction in the set to determine the Posting Code to be used. Defined codes are:

KEY General Ledger Organization Key

OBJ General Ledger Object Code

XXXX GL Key Part or Object Group code 4-character description

CKID Check Stock/Bank Code defined as CKID/@@ Common Codes

BANK Check Stock/Bank Code defined as CKID/@@ Common Codes

CNT Contribution Number (CP or EP only)

DED Deduction Number (DP only)

DIV Division Code (AP, AR, CR, or LC only)

PAY Hours Code (EP only)

M1 Employee Type (SC only)

WHSE Warehouse (SI only)

CUST Customer Type (SI only)

is between: This field allows for the entry of a specific code which relates to the IFAS structure indicated in the 'If' field.

and: Enter the second code when defining a range, or leave blank if a singular item.

then the Posting Code is: The specific posting code (001, 002, or 003, etc.) which should be used under the defined 'If ____ is between ____ and ____' conditions. Under certain circumstances it may be necessary to relate two or more 'If' conditions together. When this is the case place the word 'AND' in this field. In the last defined 'If' entry, place the required Posting Code.

Find Subsystem/Ledger: In BROWSE-mode, enter the desired two-character Ledger code and the two-character subsystem code to view the desired Posting Preferences.

****NOTE:** There is a limit of 2500 entries on the DEFAULT POSTING CODES form per Subsystem and Ledger.

The following example shows that for Accounts Payable entries (AP) coming from the Operational Ledger (GL) only accounts entered from Fund 10 through Fund 20 will be allowed and that Posting Code '01' will be automatically used.

6.3.1 Special Definitions for Payroll

In the case of Payroll, Posting Preferences may be used to group related CDH codes together so that they may be treated the same way for posting to the General Ledger. For example, if special deduction posting should occur for Electronic Funds Transfer (EFT), and you have 12 EFT Deduction Codes, it might be easier to group all 12 Deduction Codes as one special pseudo code and define GLUTSPSI around the pseudo Deduction Code. The pseudo Deduction Code defined in Posting Preferences is then placed in the 'PCi' field on the GL SUBSYSTEM INTERFACE form, GLUTSPSI, along with the special account posting information.

If the clients' Payroll GLUTSPSI does not reference the PCi field, then defining Posting Preferences is NOT NECESSARY.

For example, Deduction Codes '230' through '240' all need to be posted to a single account. We would like to group them together in Posting Preferences and identify their posting with one (1) Posting Code, PCi '023', on the GLUTSPSI Form.

6.4 What Is Going On At The System Level

The program which creates the posting entries uses the Common Codes part of NUCLEUS to store these entries. It is not important for the user of the GL SUBSYSTEM INTERFACE screen to know how these entries are stored; however, if one does understand where these entries are actually stored, it is easy to understand why certain entries exist in the Common Codes file. The following describes what the system is actually doing with the information which is placed on the GL SUBSYSTEM INTERFACE form.

Each Common Codes entry for a posting distribution has the following fields:

Code Category: A four-character field which identifies the subsystem to which the interface relates. The system enters 'lgxx' in this field, where 'lg' is the two-character Ledger code and 'xx' is the two-character code for the particular subsystem. For example the GL Ledger entries for Accounts Receivable, Cash Receipts and Payroll would be GLAR, GLCR, and GLPY respectively.

Code Value: The system makes this entry as the Report or Register name plus a four-digit number (e.g., DIST0001, DREG0001). The four-digit number is incremented by 1, for each entry that belongs to the same subsystem and the same Report/Register name.

Short Desc: This is an eight-character field which is divided into five parts: L T D PPP FF where 'L' is the level, 'T' is the type, 'D' is the posting date, 'PPP' is the Posting Code and 'FF' is the Fund Code.

Medium Desc: The title of the transaction, as it is to appear in the General Ledger. If this is 'TRNS' then the system will use the description from the transaction. Other titles are automatically created by the system and are based upon the Level, Type and PCi designations. The terminology used can be found in the Common Codes entries with a Code Category of 'GLTI' and a Code Value of the corresponding combinations of Level and Type (a P is appended if a Posting Code is given). For example:

Code Category: GLTI

Code Value: OD

Medium Desc: Object TTL; Discount Amount

Code Category: GLTI

Code Value: ODP

Medium Desc: Object TTL; Discount; PC = xxx

The user may BROWSE these entries and change their description to any desired terminology ('xxx' is replaced by the actual Posting Code).

Long Desc: Has no effect upon the resultant transaction.

Assoc Code 1: 'DR' for debit or 'CR' for credit.

Assoc Desc 1: 'CHECKS' if this entry is to be generated only when checks are printed. Any other characters in this field are unimportant.

Assoc Code 2: Holds the Org. Key designation, as provided on the GL SUBSYSTEM INTERFACE screen.

Assoc Desc 2, 3, 4 and 5: Have no effect on the resultant transaction.

Assoc Code 3: Holds the Object Code designation, as provided on the GL SUBSYSTEM INTERFACE screen.

Assoc Code 4: Hold 'POST' if the GL Posting Date is to be used as the primary transaction date in the GL; 'CHECK' if the Check Date is to be used as the primary transaction date in the GL; 'REF' if the transaction reference date is to be used as the primary transaction date in the GL.

Assoc Code 5: Hold 'REF' if the transaction reference is to be used as the primary reference in the GL transaction; 'xxxxxxxx' if the entry is in some way a summary entry.

Each Common Codes entry for the DEFAULT POSTING CODES screen definitions has the following fields:

Code Category: A four-character field which identifies the subsystem to which the interface relates. The system enters 'lgxx' in this field, where 'lg' is the two-character Ledger code and 'xx' is the two-character code for the particular subsystem. For example the GL Ledger entries for Accounts Receivable, Cash Receipts and Accounts Payable would be GLAR, GLCR, and GLAP respectively.

Code Value: The system makes this entry as the Report or Register name plus a four-digit number (e.g., DFLT0001, DFLT0002). The four-digit number is incremented by 1, for each entry that belongs to the same subsystem and the same Report/Register name.

Short Desc: This is an eight-character field which is automatically filled with the word Default for information purposes only.

Medium Desc: Has no effect upon the resultant transaction.

Long Desc: Has no effect upon the resultant transaction.

Assoc Code 1: Identifies the portion of the GL structure, Subsystem Division, etc. defined in the 'IF' field of the DEFAULT POSTING CODES form.

Assoc Desc 1: Automatically defined as the Attribute Name.

Assoc Code 2: Holds the first or low value which is to be used in determining the Posting Code.

Assoc Desc 2: Automatically defined as the Attribute Low Range.

Assoc Code 3: Holds the second or high value which is to be used in determining the Posting Code.

Assoc Desc 3: Automatically defined as the Attribute High Range.

Assoc Code 4: Holds the Posting Code which should be entered on the Set Entry Form based on the attributes defined in Assoc Codes 1 through 3.

Assoc Desc 4: Automatically defined as the Default Posting Code.

Assoc Code and Desc 5: Not used at this time.

6.5 Automatic Offset Definitions for Journal Entries

IFAS provides for six (6) different methods in creating automatic offset entries which are posted at the time a Journal Entry Set is distributed. Generally, one method will be chosen and allow the system to default for all JE Sets. Each of the methods is described below. Default values are defined in NUCLEUS Common Codes with a Code Category of 'GLJE' and a Code Value of 'OFFSET'. The Short Description defines the default value used by IFAS to automatically select the desired posting method.

The sample transactions listed below will be used in each of the following examples.

Sample Transactions for a Journal Entry

Org. Key	Object Code	DR	CR
1000	8801	10	
1000	8802	20	
2000	8801		30

Method 1

Single Object used for Due To/From for both Org. Key and Fund balancing (Intrafund).

Intrafund Object Code = 1025

What the Accountant Wants

POSTED WHEN THE JOURNAL ENTRY IS DISTRIBUTED

Dept One – Exp One (1000-8801)	10	
Dept One – Exp Two (1000-8802)	20	
Dept One – Intrafund Clearing (1000-1025)		30
Dept Two – Exp One (2000-8801)	30	

Dept Two – Intrafund Clearing (2000-1025) 30

To accomplish this posting method, enter the Intrafund Clearing Object Code on the OBJECT STRUCTURE Form (form 3) in **GLUPGN**.

To have Journal entries default to this method, enter 'YN' in the Short Description of the Common Code 'GLJE-OFFSET', **NUUPCD**.

Method 2

Single Object used only for Due To/From for Fund balancing at the Funds Administration level (Interfund).

Interfund Object Code = 1029

What the Accountant Wants

POSTED WHEN THE JOURNAL ENTRY IS DISTRIBUTED

Dept One – Gen Exp One (1000-8801) 10

Dept One – Gen Exp Two (1000-8802) 20

Gen Fund 10 – Interfund Clearing (0010-1029) 30

Dept Three – Gen Exp One (3000-8801) 30

Rest Fund 20 – Interfund Clearing (0020-1029) 30

FDID Common Codes for Fund 10 and Fund 20 are required to tell the posting process which Org Key is to be used as the Fund Control Key. In this case, 0010 and 0020 are used. A common code is required for EACH fund, in this case Fund 10 and Fund 20. The offsetting keys are specified in the first associated value, in this case 0010 and 0020. The second associated value should be left blank, as shown below.

Code Category	Code Value	Associated Code #1	Associated Code #2
FDID	10	0010	
FDID	20	0020	

To accomplish this posting method, enter the Interfund Clearing Object Code on the OBJECT STRUCTURE Form (form 3) in **GLUPGN**.

To have Journal entries default to this method, enter 'NY' in the Short Description of the Common Code 'GLJE-OFFSET', **NUUPCD**.

Method 3

Intrafund entries made for Org. Key balancing and Interfund entries made for Fund balancing.

Intrafund Object Code = 1025

Interfund Object Code = 1029

What the Accountant Wants

POSTED WHEN THE JOURNAL ENTRY IS DISTRIBUTED

Dept One – Gen Exp One (1000-8801) 10

Dept One – Gen Exp Two (1000-8802) 20

Dept One – Intrafund Clearing (1000-1025) 30

Gen Fund 10 – Intrafund Clearing (0010-1025) 30

Gen Fund 10 – Interfund Clearing (0010-1029) 30

Dept Three – Gen Exp One (3000-8801) 30

Dept Three – Intrafund Clearing (2000-1025) 30

Rest Fund 20 – Intrafund Clearing (0020-1025) 30

Rest Fund 20 – Interfund Clearing (0020-1029) 30

FDID Common Codes for Fund 10 and Fund 20 are required to tell the posting process which Org Key is to be used as the Fund Control Key. In this case, 0010 and 0020 are used. A common code is required for each fund, in this case fund 10 and fund 20, to specify to offsetting keys, in this case 0010 and 0020. Note that the keys are specified in the first associated code field. The second associated code field should be left blank.

Code Category	Code Value	Associated Code #1	Associated Code #2
FDID	10	0010	
FDID	20	0020	

To accomplish this posting method, enter the Intrafund and Interfund Clearing Object Codes on the OBJECT STRUCTURE Form (form 3) in **GLUPGN**.

To have Journal entries default to this method, enter 'YY' in the Short Description of the Common Code 'GLJE-OFFSET', **NUUPCD**.

Method 4

A single Object Code is defined for each individual fund and is posted in the **same** fund to indicate the transfer as a net Due To/From at the Funds Administration level.

Due To/From Fund 10 Object Code = 1010

Due To/From Fund 20 Object Code = 1020

What the Accountant Wants

POSTED WHEN THE JOURNAL ENTRY IS DISTRIBUTED

Dept One – Gen Exp One (1000-8801) 10

Dept One – Gen Exp Two (1000-8802) 20

Gen Fund 10 – Due To/Fr Fd 10 (0010-1010) 30

Dept Three – Gen Exp One (3000-8801) 30

Rest Fund 20 – Due To/Fr Fd 20 (0020-1020) 30

FDID Common Codes for Fund 10 and Fund 20 are required to tell the posting process which Org Key is to be used as the Fund Control Key. In this case, 0010 and 0020 are used. A common code is required for each fund, in this case fund 10 and fund 20, to specify to offsetting keys, in this case 0010 and 0020. Note that the keys are specified in the first associated code field.

Code Category	Code Value	Associated Code #1	Associated Code #2
FDID	10	0010	1010
FDID	20	0020	1020

To accomplish this posting method, **NO ENTRIES** are made on the OBJECT STRUCTURE Form (form 3) in **GLUPGN**. Instead, the Due To/From Object Codes are entered as the Second Associated Code for each fund defined in the Common Code Category of 'FDID'. For more information, refer to the discussion earlier in this section.

To have Journal entries default to this method, enter 'MA' in the Short Description of the Common Code 'GLJE-OFFSET', **NUUPCD**.

Method 5

A single Object Code is defined for each individual fund and is posted to **different** funds indicating the transfer of funds as a net Due To/From each fund at the Funds Administration level.

Due To/From Fund 10 Object Code = 1010

Due To/From Fund 20 Object Code = 1020

What the Accountant Wants

POSTED WHEN THE JOURNAL ENTRY IS DISTRIBUTED

Dept One – Gen Exp One (1000-8801) 10

Dept One – Gen Exp Two (1000-8802) 20

Gen Fund 10 – Due To/Fr Fd 20 (0010-1020) 30

Dept Three – Gen Exp One (3000-8801) 30

Rest Fund 20 – Due To/Fr Fd 10 (0020-1010) 30

FDID Common Codes for Fund 10 and Fund 20 are required to tell the posting process which Org Key is to be used as the Fund Control Key. In this case, 0010 and 0020 are used. A common code is required for each fund, in this case fund 10 and fund 20, to specify to offsetting keys, in this case 0010 and 0020. Note that the keys are specified in the first associated code field.

Code Category	Code Value	Associated Code #1	Associated Code #2
FDID	10	0010	1010
FDID	20	0020	1020

To accomplish this posting method, **NO ENTRIES** are made on the OBJECT STRUCTURE Form (form 3) in **GLUPGN**. Instead, the Due To/From Object Codes are entered as the Second Associated Code for each fund defined in the Common Code Category of 'FDID'. (See the discussion earlier in this section)

To have Journal entries default to this method, enter 'MB' in the Short Description of the Common Code 'GLJE-OFFSET', **NUUPCD**.

Method 6

A separate Object Code is defined for Due To and Due From Activity. These are posted to the appropriate fund indicating the total transfer of monies at the Funds Administration level.

Due From Fund 10 Object Code = 1010

Due From Fund 20 Object Code = 1020

Due To Fund 10 Object Code = 2010

Due To Fund 20 Object Code = 2020

What the Accountant Wants

POSTED WHEN THE JOURNAL ENTRY IS DISTRIBUTED

Dept One – Gen Exp One (1000-8801) 10

Dept One – Gen Exp Two (1000-8802) 20

Gen Fund 10 – Due To Fd 20 (0010-2020) 30

Dept Three – Gen Exp One (3000-8801) 30

Rest Fund 20 – Due From Fd 10 (0020-1010) 30

FDID Common Codes for Fund 10 and Fund 20 are required to tell the posting process which Org Key is to be used as the Fund Control Key. In this case, 0010 and 0020 are used. A common code is required for each fund, in this case fund 10 and fund 20, to specify to offsetting keys, in this case 0010 and 0020. Note that the keys are specified in the first associated code field.

Code Category	Code Value	Assoc. Code #1	Code #2	Code #3
FDID	10	0010	1010	2010
FDID	20	0020	1020	2020

To accomplish this posting method, **NO ENTRIES** are made on the OBJECT STRUCTURE Form (form 3) in **GLUPGN**. Instead, the Due From Object Code is entered as the Second Associated Code and the Due To Object Code is entered as the Third Associated Code for each fund defined in the Common Code Category of 'FDID'. (See the discussion earlier in this section)

To have Journal entries default to this method, enter 'MB' in the Short Description of the Common Code 'GLJE-OFFSET', **NUUPCD**.

Special Note for Methods 4, 5, and 6: Because of the complexity in calculating the appropriate automatic offsetting entries, care must be made in the method of creating a Journal Entry across funds. If the case should arise where multiple Debits are required from multiple funds, then separate JEID entries must be made.

6.6 Controlling Open Posting Periods

From an internal control perspective, it may be advantageous to restrict the accounting periods which are open for posting to the GLDB. Each individual subsystem, JE, AP, AR, CR, etc. asks the question, "Enter the GL Posting Date: (RETURN=MM/DD/YY):". Each question may be shown with its own default date. This is discussed in the NUCLEUS User Guide. The open accounting periods are controlled through an Edit Code for each question. The following Edit Codes are currently supported:

DT

Any valid date in the MM/DD/YYYY format. This allows all periods, past, present, and future, to be open at all times.

DC

A moving 60-day period on either side of the default posting date. If a date outside of this period is entered, the system will provide a warning which the user may override and continue the posting process.

DS

A specific period of time, as defined by the user in NUCLEUS Common Codes. Dates entered outside the specific range are blocked and the correct date range is displayed. There is not an override for this error message.

DM

Defines the number of days within the current month through which postings to the last day of the prior month will be accepted. If the current date exceeds the defined number of days, postings to the current month only will be allowed. The number of days is defined within NUCLEUS Common Codes.

The following steps are required to indicate a specific Edit Code:

1. Find the Question ID for the Subsystem Posting question. This is accomplished by going through the dialogue and when the posting question is asked, key in '??'. The system will respond with the Question ID stored in NUCLEUS.

2. Access NUCLEUS Job Running/Management by keying in **NUUPJB**. In BROWSE-mode, find the Question ID. Place the two-character Edit Code (e.g., 'DS', 'DC', etc.) on the Question ID for the specific posting and press ENTER.

The following steps are required to set up the 'DS' posting option:

1. Access NUCLEUS Common Codes by keying in **NUUPCD**. In COLLECT-mode, key in Code Category, 'NUDT'. The Code Value should be the question number (e.g., GL29 for Journal Entries). The short, medium and long descriptions are only for internal documentation and are optional. In the Associated Codes area, place the Start Date for the period on the first line and the End Date for the period on the second line. The Associated Description area may be used for internal documentation. Press ENTER and exit from NUCLEUS. The example below shows the Common Code definition for Journal Entry Distributions and identifies the quarterly period 01/01/92 through 03/31/92.

2. At the point in time when the posting period ends, the user must access Common Codes again, BROWSE the 'NUDT' entries and adjust the posting period.

The following steps are required to set up the 'DM' posting option:

1. Access NUCLEUS Common Codes by keying in **NUUPCD**. In COLLECT-mode, key in Code Category, 'NUDV'. The Code Value should be the question number (e.g., GL29 for Journal Entries). The short, medium and long descriptions are only for internal documentation and are optional. In the First Associated Codes area, place the number of days after which postings to the prior month will not be allowed. The Associated Description area may be used for internal documentation. Press ENTER and exit from NUCLEUS. The example below shows the Common Code definition for Journal Entry Distributions and identifies that postings to the prior month can be made within the first ten (10) days.

6.7 Subsystem Interface Calculations

The Type Codes in the G. L. SUBSYSTEM INTERFACE screen are default definitions. To override these defaults, complete the CALCULATION DEFINITIONS screen for the appropriate Subsystem(s) and Ledger(s). Calculation Definitions are valid in the AP, AR, CR, and SI subsystems. To define a new Calculation Definition, key in **GLUTSPSC**. The valid codes that may be used in the Calculation fields are shown on the left side of the screen. The screen as it appears in COLLECT-mode is shown below:

The CALCULATION DEFINITIONS screen below illustrates an example of how Type Codes might be defined for Accounts Payable.

The screen above illustrates how the Calculation Definitions could be defined which would allow for the posting of Tax and Discount as separate items from the Distribution and Charge amounts. The following explains the use of the Codes:

Type Code 'X' = The calculated amount of 'Distribution' plus 'Tax' plus 'Charge'. This represents the 'Total' amount of the Item from the Accounts Payable data entry screen. This is the amount that would typically be posted to the Accounts Payable control account in the General Ledger.

Type Code 'Y' = The calculated amount of 'Distribution' plus 'Charge'. This is the amount of the 'Expense' to be posted to the Account Number as entered on the Accounts Payable data entry screen.

Type Code 'Z' = The calculated amount of 'Distribution' plus 'Tax' plus 'Charge' minus 'Discount'. This is the amount that would be posted as the credit to Cash in the General Ledger.

In summary, the above Calculation Definition in conjunction with the G. L. SUBSYSTEM INTERFACE screen for Accounts Payable would cause the following postings to occur:

At the time the invoice is posted:

DR 'Y' (A + C)

DR 'T' (T)

CR 'X' (A+T+C)

At the time the check is posted:

DR 'X' (A+T+C)

CR 'D' (D)

CR 'Z' (A+T+C-D)

Shown below is the **GL SUBSYSTEM INTERFACE** screen illustrating the use of the Calculation Definition Type Codes as defined above.

7 Utility Functions

7.1 Overview

From time to time, the user may find it necessary or useful to perform special functions upon the General Ledger Database. These functions, called Utility Functions, are accessed by keying in **GLUT** from the main menu. The following question and menu will be displayed:

Which General Ledger Utility function is desired?

AC	Browse Actual Amounts	**AS**	Account Spreading Utilities
CS	Client Specific	**DI**	GL Data Integrity Utilities
RI	Rollover & Interface Utilities	**SM**	Browse Summary Accumulators
SP	Subsystem Posting	**SU**	GL Structure Utilities

Enter the two-character code which corresponds to the desired Utility Function. The following Sections will provide a more complete description of each of the General Ledger Utility functions:

7.2 Browse Actual Entries (AC)

The BROWSE ACTUAL ENTRIES screen is used to interactively view GLA-ACT-MSTR records that have been computed within the General Ledger database. The actual balances and encumbrance balances are displayed for each Object Code within the Org. Key of the appropriate Ledger and Fiscal Year in total by individual month format. If discrepancies arise while viewing these records, a General Ledger recompute should be performed (GLUTDICP). See section 8-E of this guide for more information.

If help is needed on a specific field, simply place a question mark (?) in the first position of the field and press the ENTER key. The system will display information regarding the specific field. If the field contains coded information, the system will also show valid values that may be 'picked' to go into the field. The HELP text for a specific field contains codes at the end of the description. These codes are abbreviated descriptions of what the system does with values that are placed into the field. These codes, and their meanings, are shown below:

Certain fields for which HELP has been requested hold coded information which is defined in Common Codes and in various databases throughout the system. The on-line HELP specifies how to create or update this coded information. However, it should be understood that access to some of this coded information is controlled through security. Thus, not all users of the system will have access to create or update these code values.

FY: The fiscal year of the actuals and encumbrance totals being viewed.

Ledger: The ledger code of the actuals and encumbrance totals being viewed.

Account: The ledger code of the actuals and encumbrance totals

Level: The account level of the actuals and encumbrance totals being viewed.

FP Totals: Total actuals by period for this specific FY, Ledger, Account, and Level combination.

Total: Total actuals for the fiscal year for this specific FY, Ledger, Account, and Level combination.

EN Totals: Total Encumbrances by period for this specific FY, Ledger, Account, and Level combination.

Open: Amount of Balance Forward entries included in the first fiscal period of this fiscal year for this account.

Close: Amount of Closing entries included in the last fiscal period of this fiscal year for this account.

BROWSE: Enter the Fiscal Year, Ledger, Account Number, and Level to browse.

7.3 Account Spreading Utilities (AS)

After requesting the running of Account Spreading Utilities by entering 'AS' in response to the General Ledger Utility function prompt, the system will display the following question:

Which Account Spreading Function would you like?

FL Spread Code File Listing **UP** Update Spread Codes

Note: You will actually use these codes when you go to the various update screens (APOHBTUB) and at the account number prompt you enter "+" and the code you created. When you run the set and distribution reports you will see the spread of the account numbers you specified.

Enter the two-character code which corresponds to the desired Account Spreading Utility.

Next, enter the Code you wish to use. It may be up to 6 characters in length and may be any combination of numbers and letters.

Enter the Subsystem for which this Spread Code is valid. Currently, AP, AR, PO, and PY subsystem daily time card sets are allowed in this field with spread code expansion functioning in the AP/OH subsystem only. Future updates will allow use in other subsystems.

Enter a description for the Spread Code. This description will be reflected on the File Listing report.

Enter a percentage and an account number on each line. Up to 9999 allocation lines may be entered for each Spread Code. The percentage must be less than 100%. Twenty-five percent is entered as "25.000". Leading and trailing zeros are unnecessary. Up to three decimal places of precision are used. This precision will allow the allocation of as little as $10 for a one million dollar transaction. The account number must be valid. Enter a complete account number here as the format of the Spread Code on your transaction will determine which parts of the account number are used (see below). When ENTER is pressed, the descriptions of each account's KEY and OBJECT will be echoed. Each time ENTER is pressed, the total of the percentage you have entered will be displayed in the lower right corner of the screen. Each Spread Code must have a total allocation of 100% (you will not be able to leave the current Spread Code unless it totals either 0% or 100%).

In BROWSE mode, you must type in the Spread Code and Subsystem at the bottom of the screen and then press ENTER to have that Code's data displayed. The ledger code the user is currently using will be used for all Spread Code entry and browsing. Once a Spread Code is displayed, you may use F5 and F6 to move backward and forward through the pages of data for this one Spread Code. To view a different Spread Code, type the Code and its Subsystem in the browse fields at the bottom of the screen and press ENTER.

If DELETE (F2) is pressed twice in Browse mode while viewing a Spread Code, the entire Spread Code will be deleted. To remove single lines from the Code, blank them out with the space bar.

Note: Spread codes work for Accounts Payable, but they are not designed to work for lonesome cash and cash receivable. The Common Code SYNO SPRDCODE is used to determine the file name in which the spreading information is stored for a specific spread code. When this Common Code is first set up, the first and second numeric values may be blank (zero) or '1'. The first code must be "SPRD 04", with the "SPRD" at the far left of the field and "04" at the far right. A Common Code and a file of spread code information will be generated by the system for each spread code created.

7.4 Client Specific (CS)

These are special utilities which perform a specific function for the client indicated. Clients who choose to run a Clients Specific Utility which is not written for them do so at their own risk. This is highly discouraged. The following question and menu will be displayed when this function is selected:

Which Client Specific Utility is desired?

CE Create EIMS file (Canton) **PE** PEIMS financial extract (SISD)

RK Roll Keys to Current FY (SISD)

Enter the two-character code which corresponds to the desired Client Specific Utility. The following provides a more complete description of each of the Client Specific Utilities:

CE

This utility is specifically for Canton City Schools. This utility will extract the necessary information to create the EIMS file.

PE

This utility is specifically for Spring Independent School District. This utility will extract the necessary information to create the PEIMS report.

RK

This utility is specifically for Spring Independent School District. This utility will roll Fully Qualified Org. Keys from one fiscal year to a new fiscal year. The new Fully Qualified Org. Keys will have the new fiscal year designation, while keeping all other Org. Parts the same. The system will automatically create new 'Short' Org. Keys for the new Fully Qualified Org. Keys.

7.5 GL Data Integrity Utilities (DI)

This function allows the user access to all Data Integrity Utilities. These utilities should be run if there is a suspicion that the data in the General Ledger database has been corrupted. The following question and menu will be displayed when this function is chosen:

Which Data Integrity Utility is desired?

CA Recompute Acct Bal's (UNIX)
CI Selectively change PE ID's
CP Recompute Account Balances
FE Execute RB, RL, CP
FP Verify Fiscal Per. on Trns-Dtl
PB Purge Prior Year Budgets
PU Purge Zero Budgets & Actuals
RB Rebuild Fastkey File
RL Reset the GL Link Structures
RS Verify Rule Logic on selected Chart-of-Accounts
RT Verify Logic on selected Transactions
RV Create Reversal Set
VB Verify Budgets
VC Verify GL Check Numbers
VF Verify Fastkey File
VK Verify Full Format Keys
VM Verify Monthly Budgets
VS Verify GL Structure
VT Verify Transactions

Enter the two-character code which corresponds to the desired Data Integrity Utility. The following provides a more complete description of each of the Data Integrity Utilities:

CA

This is a special version of the recompute that runs faster by not performing database locks and unlocks around writes. It is, therefore, up to the user to insure that there are no competing processes running at the time of the recompute. When CA is chosen, the questions that follow are the same as for CP (below).

CI

This utility is used to change PE ID's. This is also a good utility for combining duplicate person/entity entries into one PE ID. This only changes PE ID's in the GLDB. To change PE ID's system wide, use the Selectively change PE ID's utility in the Person/Entity subsystem (PEUTCI). The following prompts will appear when this utility is selected:

Enter current PE ID or ‘STOP’:

Respond to this prompt by entering the full PE ID that is to be changed. Entering STOP will return the user to the main menu. After entering the PE ID to be changed, the system will echo back the name associated with that particular PE ID.

Enter the new PE ID or ‘REDO’:

If the PE ID that was entered at the first prompt is the one to be changed, enter the new PE ID. If it is not, enter REDO. This utility continues prompting as above until the first prompt is responded to with STOP.

CP

For the efficient retrieval of account balance information, the system maintains month-by-month balances. If there is a critical system failure at a point in time when these balance figures are being updated/modified, the account balance information may be inconsistent with respect to the actual transactions held in the system. The following menu will be displayed:

What would you like recalculated?: ENTER=AL:

AC	Actual Balances	**AE**	Actuals & Encumbrances
AL	ALL; Budgets, Actuals, EN	**BU**	Budget Hierarchies
EN	Encumbrance Balances		

Would you like to see a report of corrected accumulators? Y/N

GLUTDICP (Recompute Account Balances) will correct the accumulators shown by GLIQBA. It needs to be run while everyone is off of the system. It can take a long time to run, so it is best to run it overnight. If the job aborts, it should be re-run as soon as possible.

When you run GLUTDICP, you will be asked whether you want to run it on budgets, actuals, EN's, or some combination. Select whichever you want to correct. If you choose to recompute Encumbrance balances in the GLDB, by choosing AL, AE, or EN, then you will be prompted with the following question "Also recompute Encumbrance balances in the ENDB?". Enter 'Y' to recompute Encumbrance balances that are held in the ENDB. These balances must be correct so that the Encumbrance balances in the GLDB may be recalculated correctly. Enter 'N' to skip the recompute of Encumbrance balances that are held in the ENDB. Entering 'N' assumes that these balances are already correct.

You will then be asked for the fiscal year. Enter the fiscal year that needs to be corrected. (If you are running the recompute for actuals and/or budgets, you can type "ALL" and fix all fiscal years at once.) If you are running the recompute for encumbrances, you must select only one year. If you need to correct encumbrances for several fiscal years, you will need to run GLUTDICP once for each year.

If GLUTDICP is being run on budgets, then answering "Y" to the question "Check budget change accumulators?" will cause the program to recompute the total budget change amount (shown on the second screen of GLBUUP) according to the amounts in the budget change details.

If GLUTDICP is being run on budgets, then answering "Y" to the question "Check monthly budgets for consistency?" will run checking on monthly budgets.

It is generally best to answer "Y" to "Check records for validity?". This will cause the program to check and warn you if the recompute found GL accumulators with a key or object code that does not exist.

If you want a report of corrected accumulators, answer "Y". Otherwise, answer "N". If you know that something has happened (an abort of a recompute, a new set-up for how data is accumulated at the control key or group level, etc.) that will cause a great deal of corrections to appear in the report, you might want to run the report out to a file.

Check the tail sheet after GLUTDICP has been run, to make sure it completed. If it shows the error message "Invalid Org Key" or "Invalid Object", which means that there was an accumulator with a key (or object) which does not exist. This may mean that other GL transactions also have this key (or object). This data needs to be corrected, and then a recompute needs to be run.

It is possible to recompute GL for only your current ledger by adding the common code GLUT RECOMPUT with the word "LEDGER" in the short description. This may be useful when recompute time is critical. You can also set this for specific ledgers by putting the ledger code in the ledger field on the common code.

If accumulation at a group level is set in the GLBUUP screen or the GLUPKY screen, and the block or warn which sets up that accumulation is taken off, that will cause GLUTDICP to make corrections to the accumulators. This frequently happens when a user needs to post data and override the budget block. When the user does this, the data posted will not be accumulated properly. If you need to post data to an account with a budget block, even though it is over budget, change the block to a warn. This will keep the accumulators updated correctly.

FE

This function will execute the Rebuild Fastkey File, Reset the GL Link Structures, and Recompute Account Balances utilities as one process. This option is made available to automatically run these Utilities in the correct order.

FP

This function will verify that the fiscal period stored on the transaction detail record is correct based upon the posting date of the transaction.

PB

This function will purge Budgets from the fiscal year and Level specified. When using FQA selection, a budget level must be specified.

PU

This utility will purge the Organization Key and Object Association (Account) for any account which has a zero budget and against which no transactions have been posted.

RB

The system maintains a file called the Fastkey File which is used to allow the rapid retrieval of information from the GL Database in a user designated order. If for some reason, such as a hardware failure, this Fastkey File has lost its integrity, the user will observe strangely ordered information on reports, when browsing the GL Database and when using the interactive INQUIRY facility. When the user observes such

strangely ordered information, a request should be made to rebuild the Fastkey File. When this utility is requested, all users should be out of the GLDB. No report is printed when this Utility Function is requested.

RL

The GLDB utilizes a database link structure which provides rapid retrieval of Key and Object related definitions. If for some reason, it is suspected that a failure has occurred in the system (e.g., all accounts do not report with an Org. Key), this utility will reestablish all links between the Org. Keys, Org. Parts, Object Codes, and Obj. Groups. No report is printed when this Utility Function is requested.

RS

This utility allows users to verify rule logic on selected chart of accounts.

RT

This utility allows users to verify rule logic on selected transactions.

RV

This utility allows users to create reversal JE sets from other JE sets in the DATA or POSTED groups. Reversal sets are record-for-record identical to the set after which they are patterned except as follows:

The amount is changed from debit to credit or vice versa so it can cancel the ones from the JE set being analyzed. The description on the transaction can optionally be changed to a string of your choice. The reference date can optionally be changed to a date of your choice.

The JE set being used for reference is not altered in any way.

VB

This utility checks Annual Budget records for errors. A report of the errors is generated. Control-Y can be used to pause or stop this utility.

VC

This utility verifies check numbers stored on the transaction detail record is the same as the check number in the Open Hold database.

VF

This utility works in much the same fashion as the Rebuild Fastkey File, except that it does not purge the file and rebuild all entries. This process only reviews the entries made and corrects invalid structure relationships. The verify may be run at the same time that other users are accessing the GLDB. No report is generated when this Utility Function is requested.

VK

This utility verifies the Input format definition for each Ledger Code and updates the processes that utilize this feature. No report is generated when this Utility Function is requested. This function is activated by creating the GLFG FQA Common Code (refer to the Common Code guide for specific setup information). If this common code does not exist then it is treated as "KEY ONLY".

VM

This utility checks Monthly Budget records for errors. A report of the errors is generated. Control-Y can be used to pause or stop this utility.

VS

This utility checks GL General Information, Org. Keys/Parts, and Object Codes/Groups for errors. A report of the errors is generated. Control-Y can be used to pause or stop this utility.

VT

This utility checks the transaction files for errors. It also optionally checks to see that for each date and/or for each job are balanced. A report of the errors is generated. Control-Y can be used to pause or stop this utility.

7.6 Rollover/Interface Utilities (RI)

This function allows the user access to all Rollover/Interface Utilities. The following question and menu will be displayed when this function is chosen:

Which Rollover/Interface Utility is desired?

AJ	Distribute Interface into GLDB	**BR**	Setfile Rollover
IJ	Convert Interface File to JE	**IN**	ADP/GL Interface
JE	JE Interfaces	**JI**	Create 7i JE set
KP	Convert to Key Parts	**KY**	Convert to Org Keys
LJ	Load Interface File into DB	**LR**	Load Rule Logic from File
OB	Convert to Object Codes	**OH**	JE Set from OH Set
OG	Convert to Object Groups	**RL**	Keys/Objects
RS	Report Transaction Errors	**RT**	Report Chart of Accounts Errors
TI	JE Table Interface	**TR**	Transactions to Flat File

Enter the two-character code which corresponds to the desired Rollover/Interface Utility. The following provides a more complete description of each of the Rollover/Interface Utilities:

AJ

This utility will execute a process which reads a previously created interface file (in one of the formats described under mask GLUTRIIJ) or database entries (from the GLJ-JEINT-DTL); converts them into an IFAS Journal Entry set file; and posts them to the GLDB. The GLJE/REDIRECT Common Code may optionally be set up to automatically redirect (at run time) invalid accounts a designated suspense account. The GLEM/REDIRECT common code may optionally be setup to automatically send (at run time) email notification of account redirection to a selected list of users.

The data items in the GLJ-JEINT-DTL are similar to the formats used by GLUTRIIJ. Use GLUTRILJ to populate this table.

BR

This utility is used to convert set files from the IFAS IV format to the IFAS VI format. This utility should be run by clients who have converted their system from IFAS IV to IFAS VI and have undistributed sets.

IJ

This utility will execute the process which converts previously created interface files into an IFAS Journal Entry set file. If the system option flag DefAct in the first screen of GLUPGN is set to "Y", default mapping will be performed on the data. To prevent this, select the optional question "Suppress default mapping" and enter "N". This mask may be used for client specific rollovers.

Creating JEINT Set Files

JE sets may be created using GLJEUB (or GLJEML if a multi-ledger set is desired), or by creating your own set file. GLUTRIIJ may be used as an interface to convert your own set files into an IFAS JE set file. This set file should be placed in the data directory. Set files used as input may be in one of three different formats: (see following pages for actual record layouts and field definitions)

The layout is the Comma Delimited JE INTERFACE file format with each field separated by commas. The fields have no minimum length. The numeric fields are character fields. Set files in this format may be created by a text editor or preferably a spreadsheet (saved in comma-delimited file format).

The mask GLUTRIIJ used to accept a csv or non-comp or comp formatted input file. This functionality has been changed to what is shown below.

GLUTRIIJ Mask only accepts input file if padded to 512 bytes in csv format.

GLUTRIIN Mask only accepts input file if padded to 282 bytes in non-comp format.

GLUTRIIC Mask only accepts input file if padded to 252 bytes in comp format.

When GLUTRIIJ is run it will ask for the name of the input file you created (the program will automatically detect the format). The next question asks for a name to give the JE set file that you wish to be created. The third question asks if you would like the set sorted or not (if sorting is desired then the set will be sorted by the first 146 characters). The forth question asks if you would like entries summarized or not (like entries will be summarized). The next question is the standard printer question. The sixth question asks whether you would like to run with default mapping turned on. The final question is the standard run as job or interactive question.

After GLUTRIIJ has completed a set proof (GLJEBP) may be run on the resulting file, or the file may be further edited using GLJEUB.

IN

This utility creates JE Set from a file generated by ADP Payroll Services.

JE

This utility is reserved for custom interfaces between IFAS and client specific external systems. The individual prompts will be dependent upon specific client needs.

7.6.1 Rollover (using GLUTRIJI)

JI This utility will look for entries in the ifgl_je_imp table and create a 7i JE set.

Table name: IFGL_JE_IMP	**Description**	
tran_id	CHAR(38)	Client specific transaction id; used as selection criteria
tran_dt	DATE	Date and Time transaction placed in this table
ifgl_set_id	CHAR(16)	JE Set ID
ifgl_set_desc	VARCHAR2(200)	Set Description
ifgl_ref	CHAR(16)	JEID
ifgl_desc	VARCHAR2(200)	JE entry description
ifgl_subs_ref	CHAR(16)	Secondary Reference
ifgl_peid	CHAR(12)	PEID
ifgl_pedb_cd	CHAR(1)	PEDB
ifgl_ck_id	CHAR(2)	CKID
ifgl_ck_no	CHAR(8)	Check Number
ifgl_date	DATE	JE Date
ifgl_account	VARCHAR2(128)	Client specific account conversion to get IFAS account
ifgl_gl_gr	CHAR(2)	GL Ledger Code
ifgl_gl_key	CHAR(10)	GL Org Key
ifgl_gl_obj	CHAR(8)	GL Object Code
ifgl_jl_gr	CHAR(2)	JL Ledger Code
ifgl_jl_key	CHAR(10)	JL Org Key

ifgl_jl_obj	CHAR(8)	JL Object Code
ifgl_wo	CHAR(12)	Work Order Number
ifgl_offset	CHAR(2)	Offset code
ifgl_misc	CHAR(4)	Misc code
ifgl_type	CHAR(2)	JE type
ifgl_units	NUMBER(20,5)	Number of units
ifgl_msur	CHAR(4)	N/A
ifgl_dr	NUMBER(20,2)	Debit amount
ifgl_cr	NUMBER(20,2)	Credit amount
ifgl_hit_en	CHAR(2)	Hit EN Flag
ifgl_en_type	CHAR(2)	EN Type
ifgl_budget_over	CHAR(2)	Budget Override Flag
ifgl_prep_id	CHAR(8)	Preparer's Id
ifgl_contract_no	CHAR(16)	Contract Number
ifgl_post_dt	DATE	Posting Date
ifgl_jeid_acg	CHAR(4)	JEID Approval Class Group
ifgl_trns_acg	CHAR(4)	Transaction Approval Class Group
ifgl_views	CHAR(4)	View Flags
ifgl_seq_no	NUMBER(5,0)	JE Sequence
system_id	CHAR(8)	Client specific
processed_dt	DATE	Initially NULL; filled in by interface after processing

When the GLUTRIJI process is run, the user is prompted for the name of the the JE Set ID to be created. If the user Enters SYSTEM the Set ID will come from the SYNO/BTCHIDJE seed.

The user will also be prompted for the JE interface template name. If an item is blank, null, or 'N/A..N/A' the lookup the appropriate value from an entry in the ifgl_trns_dtl table where trans_id='TEMPLATE xxxxxx' where xxxxxx is the Template Name the user entered.

The user will be prompted for the Transaction ID (tran_id) to be selected from ifgl_je_imp. This can be entered in 'like' format so 'AB%' will work. If 'ALL' is entered (and this should be the default) then all unprocessed transaction ID's will be processed. The process will run and select all entries with tran_id 's that match the above selection and where processed_dt is NULL. Use 'like' criteria to process the request replacing any * or @ with a %. (Unprocessed entries are those that have a NULL processed_dt.)

An exceptions only set proof will run immediately following the creation of the set. Sets without errors may then be posted manually. Sets with errors will need to handled manually by the users of the system and then posted as needed.

Once the entries are processed they will then be updated with the current date and time on the processed_dt field. An entry will also be made in the new ifgl_je_ack table that indicates that this transaction was processed.

KP

This function will load a file containing Organization Key Part data into the GLK-GRP-MSTR data set.

KY

This function will load a file containing Organization Key data into the GLK-KEY-MSTR data set.

LJ

This process will load previously created interface files into GLJE-JEINT-DTL dataset. The input file must be in one of the formats as described under mask GLUTRIIJ.

LR

Rule logic definitions may be entered using GLUPRU or by using this utility which will load user-defined comma-delimited files containing rule logic data into the database. The user will be prompted for the file containing the rule logic data. This file must have a record length of 512 bytes. The user will also be prompted for the rule logic file definition (ie. the format of the comma-delimited file). "DEFAULT" may be entered to use the following default comma-delimited format:

Column (Note: Field names correspond to those on the GLUPRU screen)

A

Ledger (at the top of the GLUPRU screen) - A two-character Ledger Code within which the Organization Part has been defined.

B

Organization Part - The four-character short description for the Org.Key, Part, Object, or Group being defined. OBTY, *OBT, or #OBT may be used to identify object type (RV, XP, AS, etc). #S1-#S8 may be used for Org.Key select codes. #M1-#M8 may be used for Org.Key miscellaneous codes.

C

Part Code - The actual code defined for the given Org.Key, Part, Select Code, Misc Code, Object, Object Type, or Group in Column B.

D

Ledger - A two-character code used to define which ledger is associated with the 'Part Code' in Column C. If this is left blank, then the Ledger code in Column A is assumed.

E

Org. Code - The four-character short description for the Part Code which as a specific relationship to the 'Organization Part' in Column B. OBTY, *OBT, or #OBT may be used to identify object type (RV, XP, AS, etc). #S1-#S2 may be used for Org.Key select codes. #M1-#M8 may be used for Org.Key miscellaneous codes.

F

Relational Operator - Defines the relationship between the 'Org. Code' in Column E and the 'Part Code' in Column G which may be associated with the given 'Organization Part' in Column B. Valid relational operators are: '=', '<', '>', '<>', '>=', '<=', or '=='. Note that '==' is the same as '=', but may be used to define defaults for new Org. Keys and Object Codes.

G

Part Code - The actual code defined for the given Org.Key, Part, Select Code, Misc Code, Object, Object Type, or Group in Column E. Ranges (two values separated by a '-') that are entered in this column will automatically be split into two entries, with the second entry having the opposite Relational Operator (Column F) from the first.

H

Logical Operator - This field is optional and is used when there exists a dual relationship between multiple 'Org. Codes' (Column E) and their 'Part Codes' (Column G). For example, when the Function can only be 'ADMN' when the Division is 'SD', the logical operator would be 'AND'. Valid logical operators are 'AND' or 'OR'.

I

Priority - Up to 8-characters which identifies the order in which the defined logic should be reviewed.

Optionally, a user-defined common code (the first 6-characters of the GLRL common code), which contains the file definition, may be entered. See the description of the GLRL/xxxxxxyy common code for information on how to setup the user-defined definition.

If yoy use the default file layout, field defaults may be defined via the common code GLRL/DEFAUL00. If you use a custom layout, the common code GLRL/xxxxxx00 may be used. See the common code descriptions for information on setting up these common codes.

A report will always be produced, which shows all entries that are being loaded in the database and identifies any problems with the data that might need to be fixed once the data has been loaded.

The process also supports a test mode, in which only the report is produced and no entries are written to the database. This is the best way to catch and correct any exceptions before actually loading the rule logic into the database.

OB

This function will load a file containing Object Code data into the GLO-OBJ-MSTR data set.

OG

This function will load a file containing Object Group data into the GLO-GRP-MSTR data.

OH

This utility interface program enables you to read an Open Hold set and create a Journal Entry set from it. It will prompt for posting date and offsetting codes.

RL

This utility converts keys, objects, org. Parts and obj. Groups from an Excel spreadsheet format. See Chart of Accounts Rollover, below.

RS

This utility produces a report on the selected transaction if it doesn't meet Rule Logic Mapping (GLUPRU).

RT

This utility produces report on the chart of accounts if it doesn't meet Rule Logic Mapping (GLUPRU).

TR

This utility will take selected transactions from a database and create a flat file in an RJE format. The flat file can then be used to create a JE set file on another database. The JE set can then be posted, placing the transactions on that database**.**

TI

This process reads ALL undistributed interface records from the GLJ-JEINT-DTL dataset, extracts them into Journal Entries Set files, and optionally posts them to GL. The GLDS/GL245C must be set up to automatically distribute created sets. The GLJE/REDIRECT Common Code may optionally be set up to automatically (at run time) redirect invalid accounts a designated suspense account. The GLEM/REDIRECT Common Code may optionally be setup to automatically send (at run time) email notification of account redirection to a selected list of users.

The data items in the GLJ-JEINT-DTL are similar to the formats used by GLUTRIIJ. Use GLUTRILJ to populate this table.

282 byte JE INTERFACE record layout

To create a JE set using this layout, the file can be created by a text editor or generated by a program. Remember to pad all fields with spaces so that they equal their fixed field size. Each record must have a length of 282 bytes.

FD Field Name IFAS Field Name Fixed Field Size

RJE-SET-ID	Set ID	16
RJE-REF	JEID (Primary Reference)	16
RJE-DESC	JE Description	30
RJE-SUBS-REF	Secondary Reference	16
RJE-PEID	PEID	12
RJE-PEDB	PEDB	2
RJE-CK-ID	Check Stock ID	2
RJE-CK-NO	Check Number	8
RJE-DATE	JE Date	8
RJE-GL-KEY	GL Org. Key	10
RJE-GL-OBJ	GL Object Code	8
RJE-JL-KEY	JL Org. Key	10
RJE-JL-OBJ	JL Object. Code	8
RJE-OFFSET	Offset method	2
RJE-MISC	Miscellaneous Code	4
RJE-GL-CODE	GL Ledger Code	2
RJE-JL-CODE	JL Ledger Code	2
RJE-TYPE	JE Transaction Type	2
RJE-UNITS	Units	18
RJE-MSUR	Units of Measure	4
RJE-DR	Debit Amount	18
RJE-CR	Credit Amount	18
RJE-HIT-EN	Hit EN Flag	1
RJE-EN-TYPE	EN Transaction Type	2

RJE-BUDGET-OVERRIDE	Budget Override Flag	1
RJE-WORK-ORDER	Work Order Number	12
RJE-PREP-ID	Prep. ID	8
RJE-CONTRACT-NO	Contract Number	16
RJE-POST-DATE	JE Posting Date*	8
RJE-JEID-ACG	JEID Approval Class Group	4
RJE-TRNS-ACG	Transaction Approval Group	4
FILLER	Reserved For Future Use	6

* This posting date should only be used if the set is posted using the Multiple Posting Date Distribution (GLJEDM). In this case, this date will become the Posting Date and RJE-DATE will become the Secondary Date after the Journal Entries have been posted.

512 byte comma delimited JE INTERFACE record layout

To create a JE set using this layout, the file should be created using a spreadsheet. The file should be saved in comma delimited format. Unused fields must use a column. Each record must have a record length of 512 bytes.

FD Field Name	Column	IFAS Field Name	Maximum Field Size
RJE-SET-ID	A	Set ID	16
RJE-REF	B	JEID (Primary Reference)	16
RJE-DESC	C	JE Description	30
RJE-SUBS-REF	D	Secondary Reference	16
RJE-PEID	E	PEID	12
RJE-PEDB	F	PEDB	1
RJE-CK-ID	G	Check Stock ID	2
RJE-CK-NO	H	Check Number	8
RJE-DATE	I	JE Date	8
RJE-GL-KEY	J	GL Org. Key	10

RJE-GL-OBJ	K	GL Object Code	8
RJE-JL-KEY	L	JL Org. Key	10
RJE-JL-OBJ	M	JL Object. Code	8
RJE-OFFSET	N	Offset method	2
RJE-MISC	O	Miscellaneous Code	4
RJE-GL-CODE	P	GL Ledger Code	2
RJE-JL-CODE	Q	JL Ledger Code	2
RJE-TYPE	R	JE Transaction Type	2
RJE-UNITS	S	Units	18
RJE-MSUR	T	Units of Measure	4
RJE-DR	U	Debit Amount	18
RJE-CR	V	Credit Amount	18
RJE-HIT-EN	W	Hit EN Flag	1
RJE-EN-TYPE	X	EN Transaction Type	2
RJE-BUDGET-OVERRIDE	Y	Budget Override Flag	1
RJE-WORK-ORDER	Z	Work Order Number	12
RJE-PREP-ID	AA	Prep. ID	8
RJE-CONTRACT-NO	AB	Contract Number	16
RJE-POST-DATE	AC	JE Posting Date*	8
RJE-JEID-ACG	AD	JEID Approval	4
RJE-TRNS-ACG	AE	Transaction Approval	4

* This posting date should only be used if the set is posted using the Multiple Posting Date Distribution (GLJEDM). In this case, this date will become the Posting Date and the JE Date will become the Secondary Date after the Journal Entries have been posted.

The following is a listing of the all the individual data elements included in the JE Interface Set File Layout with a comprehensive description of each.

RJE-SET-ID Up to 16 characters representing the Set ID or control code which uniquely identify this set of Journal Entries (e.g., JEBOBSMITH010196).

RJE-REF Up to 16 characters which indicate the primary reference for this Journal Entry.

RJE-DESC Up to 30 characters which describe the purpose or action causing the need for this Journal Entry.

RJE-SUBS-REF A Subsystem Reference Code of up to 16 characters which represents a secondary reference for this particular Journal Entry being interfaced. This element can be used to store the Purchase Order number if this is an Accounts Payable Journal Entry.

RJE-PEID Up to 12 characters which uniquely identify the Person/Entity (Vendor, Employee, or Customer) to be associated with this Journal Entry. This field may not be NULL.

RJE-PEDB Up to 2 characters which identify the Person Entity database which holds the record for this Vendor, Employee, or Customer which has incurred this Journal Entry.

RJE-CK-ID Up to 2 characters which identify the check stock ID used with this Journal Entry. Check Stock ID's must be defined in Nucleus Common Codes file under the code category CKID.

RJE-CK-NO Up to 8 characters which identify the check number which was used with this Journal Entry.

RJE-DATE Up to 8 characters which identify the secondary date of the Journal Entry, in CCYYMMDD format. REQUIRED

RJE-GL-KEY Up to 10 characters which indicate the GL Org. Key to which this Journal Entry should be associated. The Org. Key in this field must be defined in the GLK-KEY-MSTR (see GLUPKY screen). A Fully Qualified account may be used instead of an Org Key when using the comma-delimited format. If an FQA is used in place of an Org Key, the GLObject Code field should be left blank. REQUIRED

RJE-GL-OBJ Up to 8 characters which indicate the GL Object Code to which this Journal Entry should be associated. The Object Code in this field must be defined in the GLO-OBJ-MSTR (see GLUPOB screen). A Fully Qualified account may be used instead of an Org Key when using the comma-delimited format. If an FQA is used in place of an Org Key, the GLObject Code field should be left blank. REQUIRED

RJE-JL-KEY Up to 10 characters which indicate the JL Org. Key to which this Journal Entry should be associated. A Fully Qualified account may be used instead of an Org Key when using the comma-delimited format. If an FQA is used in place of an Org Key, the GLObject Code field should be left blank.

RJE-JL-OBJ Up to 8 characters which indicate the JL Object Code to which this Journal Entry should be associated. A Fully Qualified account may be used instead of an Org Key when using the comma-delimited format. If an FQA is used in place of an Org Key, the GLObject Code field should be left blank.

RJE-OFFSET A 2 character field describing which automatic offsetting entries should be made at distribution time. The first character position is the intra-offset code, the second character position is the inter-offset code. A blank offset field will be set to the last non-blank offset in this Journal Entry. Default offset values may be set through use of a NUCLEUS Common Code entry; Code Value 'GLJE', Code Category 'OFFSET'. See Section VII of the General Ledger user guide for a detailed discussion of the options available.

RJE-MISC Up to 4 characters to indicate a miscellaneous code associated with this Journal Entry. This may be used for any type of historical analysis, data extraction, or reporting which may be desired.

RJE-GL-CODE A two-character code which indicates the General Ledger to which this Journal Entry relates. REQUIRED

RJE-JL-CODE A two-character code which indicates the Job Ledger to which this Journal Entry relates. If no JL ledger is associated, enter "--" in this field.

RJE-TYPE Up to two characters representing the type of Journal Entry. Journal Entry Types include 'ST' for Standard Transaction, 'CL' Year End Closing Transaction, 'AJ' Audit Adjustment Entries, 'AC' Accrual Entry. Journal Entry will be converted with a transaction type of 'ST' unless otherwise noted.

RJE-UNITS The number of units purchased for this particular Journal Entry. This element has an implied 5 decimal places to the right of the decimal point. For example, 5 units will be placed into this element as 500000.

RJE-MSUR NOT USED.

RJE-DR The debit amount of the Journal Entry in dollars and cents. If the amount is to be a debit, the value will be converted as a positive amount and placed in this element. This element has an implied 2 decimal positions to the right of the decimal point.

RJE-CR The credit amount of the Journal Entry in dollars and cents. If the amount is to be a credit, the value will be converted as a positive amount and placed in this element. This element has an implied 2 decimal positions to the right of the decimal point.

RJE-HIT-EN If this field is set to Y or E (Exclusive ENDB posting, no GL), the transactions will be posted to EN with the given type in the field RJE-EN-TYPE. Note: In this case the PO number **must** be entered in the RJE-SUBS-REF field.

RJE-EN-TYPE This field contains a code for the type of encumbrance for this Journal Entry. Valid values for this field are EN for Encumbrance, DE for Disencumbered, PP for a Partial Payment, and FP for a Full Payment.

RJE-BUDGET-OVERRIDE If users creating sets have the required Budget Override security defined in Nucleus Database Security, a "Y" can be placed in this field to override budget blocks at data entry time allowing exit from and thus creation of an overbudget JE set. NOTE: This flag only allows creation of the set. Correction of the overbudget condition is still required before the set will post.

RJE-WORK-ORDER Up to 12 characters which identify the work order reference associated with this Journal Entry.

RJE-PREP-ID A 4 character code used to indicate the Nucleus User ID of who is responsible for the preparation of this Journal Entry.

RJE-CONTRACT-NO Up to 16 characters which identify the primary contract reference of this Journal Entry.

RJE-POST-DATE Up to 8 characters which identify the posting date (primary date) of the Journal Entry, in CCYYMMDD format. Note: This date will only be used if the JE set is distributed as a multiple-date set (GLJEDM).

RJE-JEID-ACG Up to 4-characters which define the approval class group for this JEID. Approval definitions may be updated using mask GLUPAP.

RJE-TRNS-ACG Up to 4-characters which define the approval class group for this transaction. Approval definitions may be updated using mask GLUPAP.

7.6.2 Chart of Accounts Rollover (using GLUTRIRL)

This function allows the user to create new IFAS Org Keys, Key Parts, Object Codes or Object Groups in set format. The utility will take a comma delimited file created by the user (columnar formatted files can also be used by special request) and create a new, separate file that can be loaded into IFAS. The IFAS mask is GLUTRIRL. The following question and menu will be displayed when this function is chosen:

Which file conversion utility?

KP	Convert to Key Parts	**KY**	Convert to Org Keys
OB	Convert to Object Codes	**OG**	Convert to Object Groups

Enter the two-character code which corresponds to the desired file conversion utility. The following provides a more complete description of each of the file conversion utilities:

KP This utility will create a new file with the converted Key Parts in it. The file will be in GLK-GRP-MSTR-REC format.

KY This utility will create a new file with the converted Org Keys in it. The file will be in GLK-KEY-MSTR-REC format.

OB This utility will create a new file with the converted Object Codes in it. The file will be in GLO-OBJ-MSTR-REC format.

OG This utility will create a new file with the converted Object Groups in it. The file will be in GLO-GRP-MSTR-REC format.

Most input files come from a spreadsheet. The spreadsheet column order from left to right must correspond to the field order defined in the Rollover Conversion and Interface Planner from top to bottom. In other words, if the KY menu option is being used, the left-most column on the spreadsheet should only have the value to be moved into the GL-GR field in the data set GLK-KEY-MSTR. The next column to the right should only have the value to be moved into the GLK-KEY field, column three will be moved into GLK-GRP-PART 1, column four to GLK-GRP-PART 2, etc. The other menu options work in the same fashion. Save the spreadsheet file to comma delimited format, and use a simple text editor to verify that the comma delimited field order matches the spreadsheet order (this is to make sure there were no hidden columns in the spreadsheet before it was saved as a comma delimited file). For each menu option, the input file must have fixed length records of length 420 bytes (On Unix systems, use the 'pad' command to ensure this. For example, 'pad 420 KEYINFIL' will pad the file KEYINFIL so that the utility can use it on Unix.).

If you answer 'Y' to the question "Filter out duplicates?", a file containing duplicates will be generated. It will be named DUPLICTS and will be put in the DATA directory. One original record will be in the output file you specify, but any successive records that have the same values in the key fields as the original record will be written to the DUPLICTS file. Below is the list of data sets, and the fields that need to be unique:

Each record for data set:Must be unique by these fields (concatenated):

GLK-KEY-MSTR (Org Keys) GL-GR, GLK-KEY

GLK-GRP-MSTR (Key Parts)	GL-GR, GLK-GRP-ID, GLK-GRP
GLO-OBJ-MSTR (Object Codes)	GL-GR, GLO-OBJ
GLO-GRP-MSTR (Object Groups)	GL-GR, GLO-GRP-ID, GLO-GRP

7.7 Subsystem Posting (SP)

This function allows the user access to all Subsystem Posting Utilities. The following question and menu will be displayed when this function is chosen:

Which Subsystem Posting Utility is desired?

<Updated 7.4>

CK	Bank ID/Check ID Preferences	**MA**	Change Account
PM	Posting Mapping	**PP**	Posting Preferences
SC	Subsystem Interface Calc's	**SI**	Subsystem Interface

Enter the two-character code which corresponds to the desired Subsystem Posting Utility. The following provides a more complete description of each of the Subsystem Posting Utilities:

CK

CKID Preferences (also known as Check ID's or Bank Id's) are set up in the Cash Receipts (CR) and Lonesome Cash (LC) modules. When working in any of the above subsystems, the system will pull the CKID code from the CKID Preferences Screen. When utilizing this feature, some factors are very important. The user must ensure that all ranges of Organization Keys using the GL parts, etc., are defined. If not, some gaps may appear and the system will not be able to derive a CKID. In addition, every CKID must be defined in Common Codes with a category of CKID and a code of the CKID to be used.

If a CKID code is not defined on the CKID Preferences Screen, the CKID code on the data entry screen will remain the same after ENTER has been pressed. If the CKID is left blank, the transaction will not be posted to the GL.

Certain fields for which HELP has been requested hold coded information which is defined in Common Codes and in various databases throughout the system. The on-line HELP specifies how to create or update this coded information; however, it should be understood that access to some of this coded information is controlled through security. Thus, not all users of the system will have access to create or update these code values.

MA

This utility is similar to GLUTSUMP, GLUTSUMO, and GLUTSUMC, but will change GL account/JL account & Work Order combinations in all datasets that have the GL account/JL account & Work Order fields. The process will change these combinations from one value to another, either one-by-one or via a conversion file.

One of the questions asked by this process will prompt for the subsystems to change. Enter the subsystems to be changed separated by commas, or, to change all subsystems, enter "ALL".

This utility currently will not change any entries in the NU, PE, PY, and TRIAD subsystems (even if selected). Subsystems that can be changed are AP, AR, EN, GL, PO, and SI. An optional *Test mode* is provided so that the process may be run and a report produced without changing entries in the database changes will occur. In *Test* or *non-Test modes*, a report will be generated displaying the changes that will occur once it is run in n*on-Test mode.* Processing can be stopped at any time by pressing CTRL-Y. The user will then be asked 'Continue?' Entering "Y" will continue processing. A file called PROGRESS can optionally be produced in the PERM group so that the user can monitor the progress if the utility. The new GL Account/JL Account(s) & Work Order(s) should already be defined before running this utility.

If a conversion file is used, it must be in the following format with a record length of 106.

GLUTSUMA Conversion file structure:

Old GL Key PIC X(10)

FILLER PIC X(1)

Old GL Obj PIC X(8)

FILLER PIC X(1)

Old JL Key PIC X(10)

FILLER PIC X(1)

Old JL Obj PIC X(8)

FILLER PIC X(1)

Old WO PIC X(12)

FILLER PIC X(1)

New GL Key PIC X(10)

FILLER PIC X(1)

New GL Obj PIC X(8)

FILLER PIC X(1)

New JL Key PIC X(10)

FILLER PIC X(1)

New JL Obj PIC X(8)

FILLER PIC X(1)

New WO PIC X(12)

FILLER PIC X(1)

PM This utility was previously described in Section 7, Subsystem Interfaces. Please refer to Section 7 for a detailed description.

PP This utility was previously described in Section 7, Subsystem Interfaces. Please refer to Section 7 for a detailed description.

SC This utility was previously described in Section 7, Subsystem Interfaces. Please refer to Section 7 for a detailed description.

SI This utility was previously described in Section 7, Subsystem Interfaces. Please refer to Section 7 for a detailed description.

7.8 GL Structure Utilities (SU)

This function allows the user access to all GL Structure Utilities. The following question and menu will be displayed when this function is chosen:

Which GL Structure Utility is desired?

AA	Archive Approval History	**CC**	Copy Org. Keys
CK	Search for keys	**CL**	Collapse Transaction
CO	Copy Obj Assoc. from Yr to Yr	**CP**	Copy Chart of Account files
CR	Copy Rule Logics	**MB**	Move Set: Posted to Data
MC	Change key-object combo's	**ML**	Copy/Change Ledger Code
MO	Change Object Number	**MP**	Change Org. Key Number
OB	One-sided JE Set Proof	**OJ**	Create One-sided Journal Entry
OS	Post One-sided JE	**PP**	Purge Org. Key Part
PR	Purge.Archive Rule Logics	**RS**	Change the GL Structure
UN	UNDO Transaction Distribution		

Enter the two-character code which corresponds to the desired GL Structure Utility. The following provides a more complete description of each of the GL Structure Utilities:

AA

This utility allows archival of approval history to file provided the approvals were completed as of a specific date and meet specified selection criteria. Entries selected to be archived will be purged from the database and written to BBB files (database image files) in the PERM group. Selection criteria includes Approval User ID, JEID, and 'As of' Date. The utility produces a archive approvals report that shows the following information for each approval: subsystem, JEID, Ledger, Approval Code, Approval User ID, Approval Date and Time, Set Record and Entry Number. Optionally this utility may be run in a trial mode which produces the archive approvals report only and does not delete entries from the database or create archive files.

CC: Copy utility for creating new Org. Keys & Key/Object relationships. This utility allows you to create a new set of Org. Keys with everything the same except one background part. Example: you want to create a new Location Y with the same Org. key structure as Location X. (Note: Location is a background Key Part.) You run this utility, selecting all Org. Keys that belong to Location X, copy them and replace the Location code with Y to create a new set of Org. Keys. Also, as part of this utility, include an option to copy all the Key/Object relationships of Location X and create the same relationships for Location Y. And lastly, allow an option to create a new Key/Object relationship to multiple Keys when adding a new Object code. Example: Object code Z is a new code that is valid for all High School Locations. Run this utility to create a valid relationship between Object code Z and all Keys that are marked as a High School Location. The value of the New Org. Key will be determined by the following GLFG NEWKEY common code (refer to the Common Codes guide for more information about creating this common code).

To run this utility(GLUTSUCC):

Which GL Structure Utility is desired?

AA	Archive Approval History	CC	Copy Org. Keys
CK	Copy Key Obj Associations	CL	Collapse Transactions
CO	Copy Obj Assoc. Yr to Yr	CP	Copy Chart of Accounts
CR	Copy Rules	MA	Change Account (GL/JL/WO)
MB	Move Set from Posted to Data	MC	Change Key-Obj Combination
ML	Change/Copy Ledger Code	MO	Change Object Number
MP	Change Org Key Number	MR	Massive Restructure DB's
OB	One-sided JE Set Proof	OJ	Create One-sided Journal Entry
OM	Post One-sided Multi-Date JE	OS	Post One-sided JE
PP	Purge Org. Key Part	PR	Purge Rules
RS	Change the GL Structure	UN	UNDO Transaction Distribution

UX Update Trans. from Ext. File

cc

Enter the Ledger Side (RETURN=GL)

GL GL Side - Ledger JL JL Side - Job Ledger

Enter Selection Criteria for the Org Key ;RETURN=00

00 No Criteria; Select All 01*Organization Key

02 Start Date 03 End Date

04 Copy Code 05 Old Code

06 Account Status 07 Group Code

08 Trans Traps 09 Director's Name

10 Budget Periods 11 Misc. Codes

12 Selection Codes 13 Key Types

14 Budget Checking Org Key 15*Fund Code

16*Functional Code 17*Budget Officer

18*Func ZI Project 19*ZI Project Code

20*Revenue/Grant 21*Division Code

23*Fully Qualified Org Key 24 BU Rpts

25 FS Rpts 26 SEL 3

27 SEL 4 28 LOCATION

29 SEL 6 30 SEL 7

32 Long Description 33 Short Description 15

Enter Fund Code Criteria: 101

Which Part do you want to change for the new KEY (s)?

01 FUND 02 FNCT

03 OFCR 04 FUZ

05 ZIP 06 RG

07 DIV

1

Enter the value for selected Part (Up to 8 characters): 212

Copy Key/Object Associations to the new Org. Keys? (Y/N) y

Enter the Fiscal year of the entries to be created RETURN=2003:

A summary of your job request is given below:

Current Ledger Code(s): GL / JL

General Ledger Function: Utilities

Utility Function: GL Structure Utilities

GL Structure: Copy Org. Keys

Ledger Side: GL Side - Ledger

Sel Criteria: 15

Fund Code Criteria: 101

Part to change: FUND

Value of the Part: 212

Copy KEY/OBJ Associations: Copy Key/Object Associations

Fiscal Year to Create: 2003

Estimated Run Time: Less than one minute

Job Queue: CS

Job Echo(Y/N), Code(R/I/Y/M/N/O/S) & Priority(01-12): YR08

Please stand by: Job will run interactively

Begin GL956 <1.00> TUE, SEP 24, 2002, 11:10 AM IFAS VII/7310 SP-0

KEYS Written : 1

BUDGETS Written: 3

End GL956 <1.00> TUE, SEP 24, 2002, 11:10 AM CPU = 1.6" Lg = GL

CK

This function directs the system to search the GLK-KEY-MSTR for Keys which have a COPY key established. For any keys found to have a COPY key established, all object codes associated with the COPY key through GLB-BUDG-MSTR will be associated with the current key through an entry in GLB-BUDG-MSTR with all budget values zero. The COPY-KEY field will then be right justified. This will occur if the put succeeded or if a duplicate master was encountered.

CL

To conserve on disc space and to improve processing efficiency, it is useful from time to time to have all transactions combined into a single balance forward entry. The typical installation holds approximately 2 to 3 years of transaction detail on-line. Generally, after a year-end balance forward has been completed this utility is requested. The system will conduct dialogue with the user to determine the desired begin and end dates, entry description and desired Journal Entry ID. The user is also prompted if a Set Proof Listing only is desired. If so, the balance forward entries are not posted to the GL Database; only a listing of the computed entries will be shown.

If there is a great deal of detail the system attempts to print, turn off spoolfile build by:

NUUPJB, mask GLUTSUCL, add a first line of "!FILE LP=$NULL". A Tail sheet is created. Recommend printing a report of the period to be collapsed before collapse. When the user commands the system to actually post the collapsed transaction entries to the GL Database, the system computes a net debit and credit entry for the total of all transactions for each account which are in the user entered begin and end date range. As usual, full selection criteria is available for the specification of a subset of the Chart of Accounts which is to have the collapsing function applied.

NOTE: When collapsing transactions, it is recommended that when doing only a subset of the Chart of Accounts, the utility be run for each month desired. This will prevent the GL Posting Audit Report from reporting an out of balance condition for the entire Chart of Accounts. The utility should be run for each fiscal year desired. This will allow the system to continue to provide fiscal-year comparison reporting.

PLEASE MAKE SURE THAT YOU HAVE A CURRENT BACK-UP COPY OF YOUR GENERAL LEDGER DATABASE BEFORE USING THIS UTILITY FUNCTION.

CO

This function carries the Key Obj association through to the next fiscal year. Selection parameters are provided within the dialogue to specify any or all accounts to be involved.

CP

This utility can copy Chart of Account Structure to files, to another ledger, and load from the files. The utility would get the Key, Parts, Object, and Group that was selected by using the General Ledger Standard Criteria questions and copy it to files or to another ledger or both.

GLUTSUCP - Utility to Copy GL Structure from Ledger to Ledger

This utility allows the user to copy an existing ledger structure to another ledger. This includes Keys, Key Parts, Objects, and Object Groups. Selection criteria may be used to copy any subset of source ledgers information. The destination ledger may optionally be created if it does not yet exist or optionally be updated to match the source ledger. This utility uses one source ledger but may be copied to multiple destination ledgers. The user can also copy the source ledger into a file that can be used as input to update ledgers in other databases.

Ledger structures may be mapped by using a file called GLUTxxyy where xx is the source ledger and yy is the destination ledger. This file resides in the report definition file directory and can be updated using the standard RDF editor under mask SYRDUB. If this file does not exist then the ledgers will map by matching the short 4 character names of the Key Parts and Object groups. If run interactively then the use will be given the option of creating the file and the user will be asked how the parts/groups map.

Which Function would you like to perform? (RETURN=02):

01 Copy Ledger to File 02 Copy Ledger to Ledger(s)

03 Copy to File and Ledger(s) 04 Load Ledger(s) from File

(Here the user decides what function is needed.)

(01 Will copy the selected entries from the source ledger into a file to be used later by option 04.)

(02 Copies the selected information from one ledger to others in the same database.)

(03 Does the same as 02 and 01 combined. 03 copies from ledger to ledger in the same database and save the source ledger's information in a file.)

(04 Copies the information from a file to one or more ledgers.)

Test Mode Only? (Y/N): N

(If 02,03, or 04 are selected you will be asked if this should be run in test mode only. If you answer YES then the database will not be updated. You will simply get a report of what would be updated if you had answered NO.)

****

A BACKUP of the GLDB is strongly recommended before running this process. If anything should go wrong you will need to restore the database.

****

Press RETURN to continue or Enter REDO to cancel this job:

(Just a reminder that a backup is a good idea should something go wrong while this utility is running.)

Which Source Ledger: RN

(Enter the source ledger. Only one may be entered and is check to see if it is a valid ledger code.)

Enter Selection Criteria for the Source Ledger? (RETURN=00):

00 All; KEY /Parts /OBJ /Groups 01 Criteria for Key

02 Criteria for Object 03 Criteria for KEY Parts

04 Criteria for OBJ Groups

00

(Here you decide what parts of the source ledger you wish to copy. If you want just the Object codes and groups then enter 02,04. Anything not selected will not be copied. Notice that 00 is the same as typing: 01,02,03,04)

Enter Selection Criteria for the Key ;RETURN=00

00 No Criteria; Select All 01*Key

02 Start Date 03 End Date

04 Copy Code 05 Old Code

06 Account Status 07 Group Code

08 Trans Traps 09 Director's Name

10 Budget Periods 11 Misc. Codes

12 Selection Codes 13 Key Types

14 Budget Checking Key 15*Fund

16*Department 23*Fully Qualified Key

32 Long Description 33 Short Description

00

Enter Selection Criteria for the Object ;RETURN=00

00 No Criteria; Select ALL 01 Object

02 Start Date 03 End Date

04 Valid Subsystems 05 Old Code

06 Type 07 Balance Type

08 Director's Name 09 Flags

10 Obj. Group 1 11 DR/CR

18 Audit User Number 19 Audit Job Number

20 Audit Create Date 21 Long Description

22 Short Description 23 Currency Code

24 Recalc Flag 25 Status

00

Enter Selection Criteria for the OBJ GROUPS;RETURN=00

00 No Criteria; Select All 01 Obj. Group 1

02 DR/CR

00

Enter Selection Criteria for the KEY PARTS;RETURN=00

00 No Criteria; Select All 01 Fund

02 Department

00

(The above selection criteria menus work just like normal. You may enter wild cards (@) and ranges. Only those items you select will be copied. Note that 00 selects all entries.)

Which Destination Ledger(s): R1,R2,R3,R4,R5,R6,R7,R8,R9,S1,&

Which Destination Ledger(s): S2,S3,S4,S5,S6,S7

(Enter up to 500 destination ledgers separated by commas. These are not validated. Notice the & at the end of the line allows multiple lines.)

Create Ledgers that don't already exist? (Y/N): N

(If you answer NO then any destination ledgers that don't exist will be skipped. A YES answer will create the ledger if it does not exist.)

Update destination ledger's structure to match source ledger? (Y/N): N

(For existing destination ledgers do you want the structure to be updated to match that of the source ledger. This will include all aspects of that ledger including options, budget versions, etc.)

Replace entries that already exist? (Y/N): N

(If a Key, Key Part, Object, or Object Group already exist in the destination ledger do you wish the source ledger's information to override it or not? A NO here will leave existing entries alone.)

Print All Entries? (Y/N): Y

(A YES here will print a detailed report of all Keys, Parts, Objects, and Groups that are added/updated. A NO will still give you a report with the number of entries updated/added.)

Enter the name of the file to create: RNDATA

(If copying to a file you must supply the name of a file that does not already exist on the system.)

Line Printer Copies, Name, Pri., & Option: 01LP 01

(Even if you don't elect to PRINT ALL you will get a report of the number of entries added/updated and a summary of how the ledgers mapped from one to another.)

A summary of your job request is given below:

Current Ledger Code(s): GL / JL

General Ledger Function: Utilities

Utility Function: GL Structure Utilities

GL Structure: Copy Chart of Accounts

Which Function: Copy to File and Ledger(s)

Test Mode Only?: No

BACKUP is recommended (GLDB): OK

Source Ledger: RN

Destination Ledger(s): R1,R1,R3,R4,R5,R6,R7,R8,R9,S1,

Destination Ledger(s): S2,S3,S4,S5,S6,S7

Sel Criteria: 00

Sel Criteria: 00

Selection Criteria: 00

OBJ Group Sel. Criteria: 00

KEY Part Sel. Criteria: 00

Create File Name: RNLEDGER.DATA.BSI

Replace existing entries?: YES

Print All Entries?: YES

Create Nonexistent Ledgers: NO

Update Ledger Structure: NO

LP Copies, Name, Pri, & Option: 01NULL 01

Estimated Run Time: Unknown; No data available

Job Queue: CS

Job Echo(Y/N), Code(R/I/Y/M/N/O/S) & Priority(01-12): YI08

(Note that I chose to run this interactively. The following dialog will only occur if you run interactively. If no mapping file exists for a ledger pair the default will be to match by the part/group names.)

Please stand by: Job will run interactively

Begin GL910 <1.04> THU, FEB 17, 2000, 1:24 PM IFAS6 /9902 SP-1

Reading RN...

Mapping from RN to R1...

Ledger Mapping File (GLUTRNR1.RDF.BSI) does not exist!!

Do you wish to create this file? (Y/N): Y

(This will create a file called GLUTxxyy where xx is the source ledger and yy is the destination ledger. This file resides in the report definition file directory and can be edited by SYRDUB. The default is to map by the names of the parts/groups.)

Organization Part Codes

RN Ledger Parts	R1 Ledger Parts
1 Fund	1 Fund
2 Department	2 Department

Map R1 Part FUND to RN Part (1-2, RETURN= 1):

Map R1 Part DEPT to RN Part (1-2, RETURN= 2):

Object Group Codes

RN Ledger Groups	R1 Ledger Groups
1 Obj. Group 1	1 Obj. Group 1
2 DR/CR	2 DR/CR

Map R1 Group GRP1 to RN Group (1-2, RETURN= 1):
Map R1 Group DRCR to RN Group (1-2, RETURN= 2):
Updating R1...
(This will continue through all of the destination ledgers.)
Program:gl910.cbl
Heading:GLUTSUCP
Mask GLUTSUCP@@
Questions: GLAC
GLAD
GLAE
GLAF
GLAX
GLAY
GLU@

CR

This utility can copy Rule Logics to files, to another ledger, and load from the files. The utility would get the Rule Logics that was selected by using the General Ledger Standard Criteria questions and copy it to files or to another ledger or both.

GLUTSUCR - Utiltity to Copy GL Rule Logic from Ledger to Ledger

This utility allows the user to copy existing rule logics to another ledger. Selection criteria may be used to copy any subset of rule logic information. The destination ledger may optionally be created if it does not yet exist or optionally be updated to match the source ledger. This

utility uses one source ledger but may be copied to multiple destination ledgers. The user can also copy the source ledger into a file that can be used as input to update ledgers in other databases.

Ledger strucures may be mapped by using a file called GLUTxxyy where xx is the source ledger and yy is the destination ledger. This file resides in the report definition file directory and can be updated using the standard RDF editor under mask SYRDUB. If this file does not exist then the ledgers will map by matching the short 4 character names of the Key Parts and Object groups. If run interactively then the use will be given the option of creating the file and the user will be asked how the parts/groups map.

Which Function would you like to perform? (RETURN=02):

01	*Copy Rules to Archive File*	*02*	*Copy from a Ledger to Another*
03	*Copy to File & Another Ledger*	*04*	*Load Rules from Archive File*

Select 01 to copy the selected entries from the source ledger into a file to be used later by option 04.

Select 02 to copy the selected information from one ledger to others in the same database.

Select 03 to perform both function described in 02 and 01. 03 copies from ledger to ledger in the same database and save the source ledger's information in a file.

Select 04 to copy the information from a file to one or more ledgers.

Test Mode Only? (Y/N): N

If 02,03, or 04 are selected you will be asked if this should be run in test mode only. If you answer YES then the database will not be updated. You will simply get a report of what would be updated if you had answered NO.

Note: A BACKUP of the GLDB is strongly recommended before running this process. If anything should go wrong you will need to restore the database.

Press RETURN to continue or Enter REDO to cancel this job:

Which Source Ledger: RN

Enter the source ledger. Only one may be entered and is check to see if it is a valid ledger code.

Enter Selection Criteria for Rule Logic; RETURN=00

00	*No Criteria; Select ALL*	*02*	*Rule COA Element*
02	*Rule Code Value*	*03*	*Ledger*
04	*COA Element*	*05*	*Relational Operator*
06	*Code Value*	*07*	*Logical Operator*

08 Priority *09 Start Date*

10 End Date *11 Status*

Choose the parts of the rule logics you wish to copy. If you want just the Part Code and Priority then enter 08,10. Anything not selected will not be copied. Notice that 00 is the same as typing: "01,02,03,04,05,06,07,08,09,10". You may enter wild cards (@) and ranges. Only those items you select will be copied.

Which Destination Ledger(s): R1,R2,R3,R4,R5,R6,R7,R8,R9,S1,&

Which Destination Ledger(s): S2,S3,S4,S5,S6,S7

Enter up to 500 destination ledgers separated by commas. These are not validated. Notice the & at the end of the line allows multiple lines.

Create Ledgers that don't already exist? (Y/N): N

If you answer NO then any destination ledgers that don't exist will be skipped. A YES answer will create the ledger if it does not exist.

Update destination ledger's structure to match source ledger? (Y/N): N

For existing destination ledgers do you want the rule logics to be updated to match that of the source ledger. This will include all aspects of that ledger including options, budget versions, etc.

Replace entries that already exist? (Y/N): N

If a Rule Logic already exist in the destination ledger do you wish the source ledger's information to override it or not? A NO here will leave existing entries alone.

Print All Entries? (Y/N): Y

Enter "Y" to print a detailed report of all Rule Logics that are added/updated. Enter "N" for a report with only the number of entries updated/added.

Enter the name of the file to create: RNDATA

If copying to a file you must supply the name of a file that does not already exist on the system.

Line Printer Copies, Name, Pri., & Option: 01LP 01

Even if you don't elect to PRINT ALL you will get a report of the number of entries added/updated and a summary of how the ledgers mapped from one to another.

A summary of your job request is given below:

Current Ledger Code(s): GL / JL

General Ledger Function: Utilities

Utility Function: GL Structure Utilities

GL Structure: Copy Rule Logics

Which Function: Copy to File and Ledger(s)

Test Mode Only?: Yes

BACKUP is recommended (GLDB): OK

Source Ledger: RN

Destination Ledger(s): R1,R1,R3,R4,R5,R6,R7,R8,R9,S1,

Destination Ledger(s): S2,S3,S4,S5,S6,S7

Create File Name: RNLEDGER.DATA.BSI

Replace existing entries?: YES

Print All Entries?: YES

Create Nonexistent Ledgers: NO

Update Ledger Structure: NO

LP Copies, Name, Pri, & Option: 01NULL 01

Estimated Run Time: Unknown; No data available

Job Queue: CS

Job Echo(Y/N), Code(R/I/Y/M/N/O/S) & Priority(01-12): YI08

Note: I chose to run this interactively. The following dialog will only occur if you run inter-actively.

If no mapping file exists for a ledger pair the default will be to match by the part/group names.)

Please stand by: Job will run interactively

Begin GL910 <1.08> THU, MAY 23, 2001, 1:24 PM IFAS VII/7110 SP-0

Reading RN...

Mapping from RN to R1...

Ledger Mapping File (GLUTRNR1.RDF.BSI) does not exist!!

Do you wish to create this file? (Y/N): Y

This will create a file called GLUTxxyy where xx is the source ledger and yy is the destination ledger. This file resides in the report definition file directory and can be edited by SYRDUB. The default is to map by the names of the parts/groups.

Organization Part Codes

RN Ledger Parts R1 Ledger Parts
========== ==========
1 Fund 1 Fund
2 Department 2 Department
Map R1 Part FUND to RN Part (1-2, RETURN= 1):
Map R1 Part DEPT to RN Part (1-2, RETURN= 2):
Object Group Codes
RN Ledger Groups R1 Ledger Groups
========== ==========
1 Obj. Group 1 1 Obj. Group 1
2 DR/CR 2 DR/CR

Map R1 Group GRP1 to RN Group (1-2, RETURN= 1):
Map R1 Group DRCR to RN Group (1-2, RETURN= 2):
Updating R1...
(This will continue through all of the destination ledgers.)
Program:gl910.cbl
Heading:GLUTSUCR
Mask GLUTSUCR@@
Questions: GLBS
GLAD
GLAE
GLBT
GLU@

PLEASE MAKE SURE THAT YOU HAVE A CURRENT BACK-UP COPY OF YOUR GENERAL LEDGER DATABASE BEFORE USING THIS UTILITY FUNCTION

MB

After the GL Utility 'UNDO' is performed (described below), it may be necessary to move the Set file of entries back into the DATA group for adjustment and correction. The system will prompt for the name of the set file. This utility will run interactively and immediately move the desired set.

MC

This function will change key-object combination(s) in ALL subsystems excluding flat files. This utility is similar to GLUTSUMO and GLUTSUMP. It will change combinations from one value to another based on the GL ledger associated with the user. The utility will prompt for the subsystems to change. Enter the subsystems to be changed separated by commas, or, to change all subsystems, enter “ALL” (subsystems that can be changed: GL,AP,EN,PY,AR,PO,SI,PE,NU) (Triad can also be changed if applicable.). This utility can be run in “Test” mode so no database changes will occur. In “Test’ or “non-Test” modes, a report will be generated displaying the changes that will occur once it is run in “non-Test” mode. Processing can be stopped at any time by pressing CTRL-Y. The user will then be asked **Continue?** Press “Y” to continue processing. A file called PROGRESS can be produced in the PERM group so that the user can monitor the progress if the utility.

If multiple key-obj combinations need to be changed, create a conversion file with the old key in the first ten columns, followed by 1 blank space, then the old object in the next 8 columns (columns 12 through 19), followed by 4 blank spaces, then the new key in the next 10 columns (columns 24 through 33), followed by 1 blank space, then the new object in the next 8 columns (columns 35 through 42). The old and new value can be shorter than ten characters. Upload the file to the DATA directory in ASCII format so that no tabs or unnecessary control characters are included in the file. Rename the file according to IFAS standards, then pad the file. For example, if the name of the file is **acctconv.txt**, rename the file to **ACCTCONV**, then pad it with the command **pad ACCTCONV** (“pad” defaults to a record length of 80, which is the required record length for the file). Once the utility has finished, run GLUTDIRL to rebuild the links between keys and key parts, objects and object groups. You must also run GLUTDICP (GL recompute).

Below is the list of tables and column names changed by GLUTSUMC, MO and MP (for Triad users: run GLUTSUMC in test mode and check the table "triadacctcomb" for column and table names):

Table	Columns
GLK-KEY-MSTR	GLK-KEY
GLK-LINK-DTL	GLK-KEY
GLO-OBJ-MSTR	GLO-OBJ
GLO-LINK-DTL	GLO-OBJ
GLBA-BUDACT-MSTR	GLBA-KEY, -OBJ
GLM-MO-MSTR	GLM-KEY, -OBJ
GLC-BUDG-DTL	GLC-FROM-KEY, -OBJ

GLQ-QUICK-MSTR	GLQ-KEY, -OBJ
GLF-FULL-MSTR	GLF-KEY, -OBJ
GLT-TRNS-DTL	GLK-KEY, GLO-OBJ
GLD-DEF-DTL	GLD-MAP-GL-KEY,
	-OBJ, GLD-GL-KEY, -OBJ
GLG-GEN-MSTR*	GLG-INTRA
GLR-RULE-DTL*	GLR-VAL
GLK-KEY-MSTR	GLK-OVER-KEY
OH-DTL	OH-GL-KEY, -OBJ
OHB-SET-DTL	OHB-GL-KEY, -OBJ
OHC-CLASS-DTL*	OHC-ADD-QUAL
EN-DTL	EN-GL-KEY, -OBJ
ENT-TTL-DTL*	ENTF-KEY, -OBJ
POI-ITEM-DTL	POI-JL-KEY, -OBJ
PON-EN-DTL	PON-GL-KEY, -OBJ
POC-CLASS-DTL*	POC-ADD-QUAL
SIT-TRANS-DTL	SIT-GL-KEY, -OBJ
SIM-ITEM-DTL	SIM-GL-KEY, -OBJ
SI-WHSE-MSTR	SI-WHSE-GL-KEY, -OBJ
SIC-CLASS-DTL*	SIC-ADD-QUAL
CD-CODES-MSTR	CD-ASSOC-CODE(1), (2)
US-DB-DTL*	US-DB-SEL-CD1, -CD2
PYN-NUM-DTL	PY-CHR-CD
PYM-CDH-DTL*	PYM-KEY(1), (2)
	PYM-OBJ(1), (2)

AR-TRNS-DTL	AR-GL-KEY, -OBJ
AR-CODE-MSTR	ARC-PROJ, ARC-OBJ,
	ARC-DIV-PROJ, ARC-DIV-OBJ
PE-PROD-MSTR*	PE-OBJ, PE-OBJ2

* Skipped by GLUTSUMC

Optionally, this process may will only change detail transactions for a selected date range. When this option is chosen, only entries in the AR-TRNS-DTL, EN-DTL, GLT-TRNS-DTL, GLA-TRNS-DTL, OH-DTL, OHB-SET-DTL, and SIT-TRANS-DTL with a posting date in the entered date range will be updated. All other datasets will not be changed. This option is useful to change current year entries, but leave prior-year entries with their existing account numbers for reporting and audit purposes.

ML GLUTSUML Copy/Change Ledger Code Utility

A full backup should be performed before running this utility in any mode.

This utility has the following options:

00 Test Mode Runs like Change Mode but produces a report only. No changes are made.

01 Change Mode Copies one Ledger Code to another (same side GL/JL) Ledger Code. All data under the old Ledger Code is deleted.

02 Copy Mode Copies one Ledger Code to another (same side GL/JL) Ledger Code. All data under the old Ledger Code remains unchanged.

03 Copy Chart of Accounts Copies the Account structure for one Ledger Code to another (same side GL/JL) Ledger Code. All data under the old Ledger Code remains unchanged. The Account Structure consists of Org. Keys, Org. Parts, Object Codes, and Object Groups.

04 Delete Mode Deletes all data for one Ledger Code. This option is used to clear Ledger Codes that are no longer used. This option is not recommended for use on Production Accounts.

Changes are made by this utility to the following Databases: GL, OH, EN, BD, PY, PP, AR, PO, SI, WO, and ROOTDB. Datasets affected by this utility (not including automatic masters): GLK-KEY-MSTR, GLK-GRP-MSTR, GLK-LINK-DTL, GLO-OBJ-MSTR, GLO-GRP-MSTR, GLO-LINK-DTL, GLF-FULL-MSTR, GLQ-QUICK-MSTR, GLD-DEF-DTL, GLR-RULE-DTL, FC-COUNTRY-MSTR, FC-CURR-MSTR, FC-CONV-MSTR, FC-RATE-MSTR, GLB-BUDG-MSTR, GLC-BUDG-DTL, GLT-TRNS-DTL, GLA-ACT-MSTR, OH-DTL, EN-DTL, ENT-TTL-DTL, ENA-ACCT-MSTR, HR-PE-MSTR, HR-REC-DTL, PYN-NUM-DTL, PY-KEY-MSTR, PYP-PAY-DTL, PYH-HIST-DTL, AR-ACTERM-MSTR, AR-ACCT-MSTR, ARPA-LINK-DTL, ARAT-LINK-DTL, AR-BAD-PAY-DTL, AR-CALC-DTL, ARMP-LINK-DTL, AR-TRNS-DTL, AR-TTL-DTL, AR-GEN-MSTR, AR-DEF-MSTR, AR-ARAR-MSTR, AR-SELC-MSTR, AR-CODE-MSTR, POI-ITEM-DTL, PON-EN-DTL, SIT-TRANS-DTL, SIM-ITEM-DTL, WO-MSTR, US-DB-DTL, CD-CODES-MSTR.

WARNING: Set files, Ad Hocs, RDF's, and TRIAD databases are NOT modified by this utility.

MO

This function will change a single Object Code or multiple Object Codes to another value(s) in ALL subsystems based on the ledger code associated with the user. If multiple objects need to be changed, an account conversion file must be made. The old object must be in the first 10 columns. A blank space must follow the first 10 columns in column 11. Then the new object should be in columns 12 through 21. The ASCII file must then be uploaded (Transfer type must be ASCII—not binary) to the DATA group. Once the file is uploaded it must be padded with the command “pad 80 ACCTCONV”(where ACCTCONV is the name of your account conversion file). Use the same procedure for GLUTSUMP. This utility also has Control-Y capability. This means processing can be paused or stopped if Control-Y is pressed. This functionality only applies to subsystems other than Triad (TR). This utility will merge objects if a new object already exists on the system (GLUTSUMP has the same capabilities).

Optionally, this process may will only change detail transactions for a selected date range. When this option is chosen, only entries in the AR-TRNS-DTL, EN-DTL, GLT-TRNS-DTL, GLA-TRNS-DTL, OH-DTL, OHB-SET-DTL, and SIT-TRANS-DTL with a posting date in the entered date range will be updated. All other datasets will not be changed. This option is useful to change current year entries, but leave prior-year entries with their existing account numbers for reporting and audit purposes.

MP

This function will change a single Org. Key or multiple Org. Keys to another value(s) in ALL subsystems based on the ledger code associated with the user. See instructions for the "MO” menu option (ABOVE).

Optionally, this process may will only change detail transactions for a selected date range. When this option is chosen, only entries in the AR-TRNS-DTL, EN-DTL, GLT-TRNS-DTL, GLA-TRNS-DTL, OH-DTL, OHB-SET-DTL, and SIT-TRANS-DTL with a posting date in the entered date range will be updated. All other datasets will not be changed. This option is useful to change current year entries, but leave prior-year entries with their existing account numbers for reporting and audit purposes.

OB

This function runs a one-sided JE set proof. Normally, GLUTSUOB will display out-of-balance warning messages. To prevent these messages from displaying, add the common code GLJE 450SUPPR (leave the fields blank; "OFF" in the short Description will turn this common code off).

OJ

The standard General Ledger Journal Entry function does not accept out of balance entries. In some instances, it is necessary to create an out of balance Journal Entry set file. This utility processes in the same manner as the standard Journal Entry Set process; however, no balance checking is performed. Normally, GLUTSUOB will display out-of-balance warning messages. To prevent these messages from displaying, add the common code GLJE 450SUPPR (leave the fields blank; "OFF" in the short description will turn this common code off).

OS

The standard General Ledger Journal Entry Posting function will not post out of balance entries. In some instances, it is necessary to post an out of balance Journal Entry set file. This utility processes in the same manner as the standard Journal Entry Set Posting process; however, no balance checking is performed.

PP

This function will purge an unwanted Org. Key Part from all Org. Keys. In addition it will remove the Part from all screens and menus and purge all values for the Part. Remaining Parts will move up in Part Code order.

Please make sure that you have a current back-up copy of your general ledger database before using this utility function.

PR

This utility purges Rule Logic entries and will ask if needed to archive the selected Rule Logics to files before purging the entries. The utility would get the Rule Logics that was selected by using the General Ledger Standard Criteria questions and PURGE/ARCHIVE the entries.

Test Mode Only? (Y/N): N

A BACKUP of the GLDB is strongly recommended before running this process. If anything should go wrong you will need to restore the database.

Press RETURN to continue or Enter REDO to cancel this job:

Just a reminder that a backup is a good idea should something go wrong while this utility is running.

Enter Selection Criteria for Rule Logic; RETURN=00

00 No Criteria; Select ALL *01 Defined Ledger*

02 Defined Org. Part *03 Defined Part Code*

04 Ledger *05 Organization Code*

06 First Relational Operator *07 Second Relational Operator*

08 Part Code *09 Logical Operator*

10 Priority

Decide what parts of the rule logics you wish to purge/archive. If you want just the Part Code and Priority then enter 08,10. Anything not selected will not be copied.(Notice that 00 is the same as typing: 01,02,03,04,05,06,07,08,09,10)

The above selection criteria menus work just like normal. You may enter wild cards (@) and ranges. Only those items you select will be purged/archived. Note that 00 will give you a warning, because assumes user selected all by mistake.

Are you archiving the selected rule logics?(Y/N): Y

Enter the name of the file to create: RNDATA

If copying to a file you must supply the name of a file that does not already exist on the system.

Print All Entries? (Y/N): Y

A YES here will print a detailed report of all Rule Logics that are purged. A NO will still give you a report with the number of entries archived.

Line Printer Copies, Name, Pri., & Option: 01LP 01

Even if you don't elect to PRINT ALL you will get a report of the number of entries added/ updated and a summary of how the ledgers mapped from one to another.)

A summary of your job request is given below:

Current Ledger Code(s): GL / JL

General Ledger Function: Utilities

Utility Function: GL Structure Utilities

GL Structure: Purge/Archive Rule Logics

Which Function: Copy to File and Ledger(s)

Test Mode Only?: Yes

BACKUP is recommended (GLDB): OK

Selection Criteria: 08,10

Archiving Rule Logic: Archive

Create File Name: RNLEDGER.DATA.BSI

Replace existing entries?: YES

Print All Entries?: YES

LP Copies, Name, Pri, & Option: 01NULL 01

Estimated Run Time: Unknown; No data available

Job Queue: CS

Job Echo(Y/N), Code(R/I/Y/M/N/O/S) & Priority(01-12): YI08

(Note that I chose to run this interactively. The following dialog will only occur if you run interactively.

Please stand by: Job will run interactively

Begin GL910 <1.09> THU, MAY 23, 2001, 1:24 PM IFAS VII/7110 SP-0

Reading RN...

(This will continue through all of the selected rule logics.)

Program:gl910.cbl

Heading:GLUTSUCR

Mask GLUTSUCR@@

Questions: GLBS

GLBU

GLU@

PLEASE MAKE SURE THAT YOU HAVE A CURRENT BACK-UP COPY OF YOUR GENERAL LEDGER DATABASE BEFORE USING THIS UTILITY FUNCTION.

RS

From time to time, the Controller may wish to alter the General Ledger structure. Before running this Utility Function, the new structure should be entered as described in the section on directly updating the GL Database, General Information. Key & object parts that already have values may be initialized by setting the short description of common code category GLDB, code value INITIAL, to YES. After specifying the new GL structure, this Utility Function should be run to automatically restructure the GL Database. When this function is requested, the GENERAL LEDGER RESTRUCTURE; screen will appear on the terminal.

Please make sure that you have a current back-up copy of your general ledger database before using this utility function.

Each set of transactions which has been posted to the General Ledger Database has a unique Job Number associated with it. To determine the Job Number associated with a particular set of transactions, the user may request a File Listing of transactions or examine the Posting Report of the subsystem from which the transactions originated. This Utility function is used to purge all transactions with a given Job Number from the GL Database.

When this function is requested, the system will prompt the user to enter the Job Number; then, all transactions with the given Job Number are purged from the GL Database and an 'UNDO Report' is printed showing all transactions purged. A sample of this report is shown in Appendix A of this user guide.

After this screen has been filled in and the ENTER key is pressed, the system will take considerable time to restructure the GL Database as prescribed on the screen.

8 Appendix A: Troubleshooting

8.1 General How To's

WE NEED TO RESTRICT AN ORG. KEY TO SPECIFIC OBJECT CODES. - When defining an Org. Key, GLUPKY, place a 'Y' in the 'Require Association' field. This means that at data entry time only those Object Codes which have been previously associated with the Key may be used.

WE NEED TO ALLOW CERTAIN CLASSIFICATIONS OF OBJECT CODES FOR A GIVEN ORG. KEY. - When defining an Org. Key, GLUPKY, you may indicate the 'Valid Object Types' which can be used with a specific Key. For instance, indicating 'RV' and 'XP' would allow only Revenue and Expense Objects to be associated with the Key.

AN ORG. KEY NEEDS TO BE TEMPORARILY INACTIVATED FROM RECEIVING ANY POSTINGS OR BUDGETS. - In the Org. Key definition form, GLUPKY, place an 'I' in the 'Status' field.

AN ORG. KEY WILL NO LONGER BE USED AFTER A GIVEN DATE. - Use the Date Range field on the Org. Key definition form, GLUPKY. This may also be used to indicate the starting date for a specific Org. Key.

WE NEED TO GROUP ORG. KEYS TOGETHER FOR A SPECIAL STUDY BUT DO NOT WANT TO CREATE ANOTHER FORMAL ORGANIZATION PART. - Use either the Select Code or the Miscellaneous Code fields for the Org. Key definition, GLUPKY. Select Codes are validated (see Section 2 for the Common Code) and Miscellaneous Codes are not.

WE NEED TO RESTRICT AN OBJECT CODE TO SPECIFIC ORG. KEYS. FOR INSTANCE, BALANCE SHEET OBJECTS ARE ONLY USED WITH FUND ADMINISTRATION ORG. KEYS. - When defining an Object Code, GLUPOB, you may place a 'Y' in the 'Require Association' field. This means that it must be previously associated with the Org. Key prior to allowing postings at data entry time. The association is made using the Budget Update process. Remember a budget may be zero, '0'.

AN OBJECT CODE WILL NO LONGER BE USED AFTER A GIVEN DATE. - Use the Date Range field on the Object definition form, GLUPOB. This may also be used to indicate the starting date for a specific Object Code.

POSTINGS FROM INAPPROPRIATE SUBSYSTEMS ARE BEING MADE TO AN OBJECT CODE. - In the Object definition form, GLUPOB, you may selectively stop various subsystems from posting to an Object Code by placing an 'N' under the two-character subsystem code.

SOME ACCOUNTS ARE USED ON A VERY FREQUENT BASIS. WE WOULD LIKE TO SHORTEN THE DATA ENTRY NEEDED FOR THESE ACCOUNTS. - IFAS provides the ability to define a 'Quick Number'. This may be a number or alpha characters which represent a single Org. Key and Object combination. For instance, Fee Revenues are always posted to '1012004001'. This can be shortened to 'FEE'. Refer to Section 2.C.1 for assistance in defining Quick Numbers.

WE NEED TO DISENCUMBER ACTIVITY OUTSIDE OF THE ACCOUNTS PAYABLE PROCESS. - Journal Entries allow the user to Disencumber a PO or other Encumbrance Reference Number which has been previously entered in the Encumbrance System. For this to occur, the 'Hit EN?' field must have a 'Y' and the 'Sec Ref' field must contain the PO or Encumbrance Reference Number. The 'Tp' field allows the user to indicate whether this transaction is to be considered a Partial 'P' or Final 'F' disencumbrance.

THROUGHOUT THE MONTH THERE ARE STANDARD JOURNAL ENTRIES WHICH NEED TO BE MADE. THE SAME ACCOUNTS ARE INVOLVED BUT THE AMOUNTS AND DESCRIPTIONS CHANGE. - Refer to Section 3.B. A master set of Journal Entries may be created and then copied using an IFAS utility. The copied file may be BROWSED and changed as needed.

THE BALANCE SHEET AND/OR BALANCE SHEET BY FUND ARE OUT OF BALANCE WITH THE GENERAL LEDGER AND DETAILED TRIAL BALANCE REPORTS. - The Balance Sheet and Balance Sheet by Fund are controlled through Report Definition Files, RDF's. It is possible that additions or changes have been made to the Chart of Accounts which alter the way in which the Balance Sheet is created. Review the RDF's in question.

THE ENDING FUND BALANCE IN THE STATEMENT OF CHANGES IN FUND BALANCE DOES NOT AGREE WITH THE FUND BALANCE IN THE BALANCE SHEET. - The Statement of Changes in Fund Balance is controlled through Report Definition Files, RDF's. It is possible that additions or changes have been made to the Chart of Accounts which alter the way in which the Statement is created. Review the RDF in question. For more information, refer to Section 5.D.

8.2 GL Out of Balance; Running GLREFLTR

Although there are many controls within IFAS to prevent the General Ledger from getting out of balance, some circumstances may occur which will cause this to happen. The user may not be aware of this condition until after a Recompute Balances utility has been run and the Audit Report at the end of a distribution indicates the condition exists.

Once the user has determined that the General Ledger is out of balance the Transaction File Listing can be run to find the specific Job which caused the out of balance condition. The file listing should be run as follows:

Mask to access file listing; GLREFLTR.

Generally use no Selection Criteria. If condition is definitely confined to one Fund, then Selection by Fund maybe appropriate.

If, per the Audit Report, the condition is confined to one particular month, restrict the Date Range to that month.

Select the Exceptions Only (EO) Report Format. The exception is a job or jobs that are out of balance. This format will only include on the report those Jobs which do not net to zero.

Sort the report by Job Number.

Total by Sort Level 1 (Job Number).

Page Break at Sort Level 1 (Job Number).

After running the report and reviewing the output, research of the transactions identified will reveal the cause of the out of balance condition.

8.3 Subsystem Is Not Deriving Posting Codes

The system will be unable to derive a Posting Code if the criteria entered on the Posting Preferences screen has not been met by some element of the transaction being entered.

The variables used in the If-Then statements on the Posting Preferences screen are what the system uses to derive a posting code at the time of data entry. These variables are the Organization Keys, Organization Part Codes, Object Codes, Object Group Codes CKID's, and AP and AR Division Codes as defined in the system.

Refer to the Posting Preference section of this guide for more information.

8.4 Errors

1a) FOUND A TOTAL LINE NAME, BUT THERE IS NO TOTAL LINE WITH THAT NAME. - The total line name it is referring to is printed above this error in capital letters. First verify that it is spelled correctly on the Total Line Definition screen. Compare the screen where this name is defined with the calculation line description (DETAIL[xxxxxx] to verify the two are spelled the same. Check all other Total Line screens for the same spelling of this name. The error message itself, when it provides the name, could be the clue you need. If you have created a total line named OBJECT and the message states OJBECT, transpositions may be overlooked.

1b) FOUND A TOTAL LINE NAME, BUT THERE IS NO TOTAL LINE WITH THAT NAME. - This occurs on those occasions where you are sorting by some grouping and ask for a total of that group. A classic example is to total by Revenue and then by Expenses. If there are no values to total within the first selected item this error will appear. For example, the error will appear when asking for the total Revenue by Account Key. If the first Account Key has no Revenue, no total line named REVENUE will be created. The second Account Key is sorted and it HAS Revenue, it will not have a Total Line named REVENUE to sum these values. The first Account Key MUST meet the full criteria of the Total Lines - it MUST have Revenues and Expenses if you intend to total by those items. Run the report again and select an Account Key that has both to verify if this is the cause. This is also associated with the popular AFTER command [(AFTER REVENUE)] for Total Line Locations. If no total line is created for revenue (REVENUE), it cannot place it AFTER.

2) UNABLE TO DETERMINE A LINE NAME ADDRESS xxxxxx - This is related to the above errors. After you have launched your job, whether to the printer or to the screen, the Table Formatter program recognizes the error because it is trying to arrange the report in the format requested. The correction is the same as above.

3a) NO RECORDS SELECTED - Verify that valid Funds, Functions, Account Keys, etc. are requested and that the date entered as a Report Date is correct. If it calls for the date of the report and the default is 07/31/91 and you have entered 07/31/1991, the program will attempt to find data for the period ending 07/31/19 - the year 1919.

3b) NO RECORDS SELECTED - If the selection at run-time seems to be correct, or you have taken all of the defaults for GL parts and dates, etc., then the problem may lie within the report itself. This assumes that there is valid data in the General Ledger for that period and those accounts.

A. Check the Item Range Definition Screen for valid dates (FYB+1 - FYE-1 could cause a problem).

B. While at this same screen, check the Special Codes box to see if your selection of RV or XP, FUND, BC, etc., are within the range of the selection made at run-time. If Account Number 10000 is not within your selected Fund 20, and this was your selection - this error would result.

C. Double check all of your Item Numbers with those available at the bottom of the screen. When adding or deleting your selections, these numbers change and must be updated every time.

D. Another problem may be due to the selection in the first three boxes where we usually put RV and XP for selection. Nothing will be selected if the user had defined an asset account in the special Selection immediately below. As the program only selects RVs and XPs in accordance with the boxes, it will never find an Object Code defined as AS - or even LI, FB, or any other of the codes.

4) UNABLE TO PARSE EXPRESSION - This error usually appears when some sort of text is entered in a calculation field. The error is saying that it doesn't know what you mean. An easy example would be entering the letter O for an item number rather than the numeral 0 (o1 instead of 01). Another example would be in the Date Range, entering BOY for Beginning of Year rather than FYB for Fiscal Year Beginning. BOY is not in the program's "dictionary" of acceptable definitions and meanings - hence it cannot parse. In other cases, adding parenthesis or modifying your calculations will help. A common 'unable to parse' expression is an attempt to multiply by .1 or .3. Example C01*.05 would give the error because it isn't expecting to multiply by a period. Add a leading zero to the formula to appear as C01*0.05 will provide a 5% calculation.

In regard to using symbolic dates (-0/01/-0), imagine the span -1/01/-1--1/31/-1 (the preceding month for the prior year). Normally, we would define the statement with parenthesis, i.e., (-1/01/-1)-(-1/31/-1), to get away from the double minus signs. The use of the parenthesis is not necessary as the program recognizes that the first of the two minus signs is translated as a "through" designator (a hyphen sign, not a minus sign).

8.5 How To’s

1) TEXT SEEMS TO REPEAT ITSELF - When the report is presented on paper or on the screen and it looks as though the same information is being printed twice on following lines:

1600 Furniture & Equipment

1600 Furniture & Equipment

Furniture & Equipment 55.00

1609 Accumulated Depreciation

1609 Accumulated Depreciation

Accumulated Depreciation (17.00)

The first place to look would be at the Sort Screen. The Lead-In for the Sort Level is printing the first line (Object Code and Object Code Description) and is probably shown as \01 \02 in that area. Because the same pattern appears again, check the Item Selection screen (#2) to determine if you requested these items with the designator of 'A' and 'B'. This will also cause the doubling up of titles. The Total Line Definition is printing the Object Description (\02 in the description field) and then following along with the balance in that account. Several Rules of Thumb:

A. Never use a Sort Lead-in at the lowest level of the report hierarchy!

B. Never use Item Designators 'A', 'B', etc. and create a Total Line for the items! 'A', 'B', etc. will print those items (in that order) within the "detail" area of the report. Sort Lead-ins (\01 - \02) will print the codes and descriptions in the "detail" area of the report. Total Line Descriptions will print this same information (Total \01 - \02) in the "detail" area of the report. Make your selection from the above methods.

2) CREATING A BLANK COLUMN - A blank column may be created just like any other column; giving it a number (in the top two-character field), not defining a Title (leave it blank) and then select for a calculation something that doesn't exist from the items listed at the bottom of the screen. Item 28 (I28) is preferred as it is least likely to be used (the program will not accept Item Numbers greater than 20 as they are for display within the detail area of the report - designated as 'A', 'B', etc.). The final step is to designate the column as "N" for Numeric but this probably isn't as critical or necessary at all. Now, when prompted if you have any blank columns, you may see the questions that hide behind that menu selection (i.e., How Wide? etc.). If it isn't going to be greater than 9 spaces, forget the trouble of it all, just designate spacing on the next column to be 9 or less to accomplish the same thing.

3) SUPPRESSING NUMBERS (DETAIL) WHERE ONLY TOTALS ARE DESIRED - In the two examples following (Report ONE and Report TWO) the meaning of this capability is shown.

Report ONE			**Report TWO**		
Budget	Actual	Variance	Budget	Actual	Variance
50.00	25.00	25.00	50.00	25.00	25.00
35.00	17.00	18.00	35.00	17.00	18.00
19.00		19.00	19.00		19.00
65.00	14.00	51.00	65.00	14.00	51.00
-----	-----	-----	-----	-----	-----
169.00	56.00	113.00	169.00	56.00	113.00

This is probably not a meaningful report, but it does show an example of streamlining a document by getting rid of the clutter. This is a report showing the balance remaining by a group rather than a line-by-line accounting. The same could be accomplished if calculating the Average of the items listed - Column 1 could be the 1st Month Total, Column 2 would be the 2nd Month Total, etc. and the Variance total (above) would be the Average of all listed (try that on your Cash Accounts for Investment average balances per period).

The secret in this style of report is in the Total Line Definition Screen. The 20 boxes that are situated near the center of the screen - where we usually insert C's for Calculation type columns. In this case, we will insert the letter B for Blank. This directs the program not to print any total for this group in that particular column. Looking at the example, a B was placed in column space 1 and 2 for the Variance Total Line. A B was placed in column space 3 for the detail totaling line (assuming a total line was created for those items) and the results are as above. In calculating the Average, place an "M" in the appropriate column box (M = Mean).

4) LISTING EXPENSES ONLY AS NEGATIVE NUMBERS IN A COLUMN - Liability and Revenue balances may be shown as negative numbers within our reports. Responding 'Y' to the prompt "Do you wish Credit Accounts to be shown as Negative numbers?" causes these appropriate balances to appear as negatives. Revenue and Expense accounts may be selected to be printed and within the combination, negative Revenues or negative Expenses may be printed. Remember those three boxes (out of the 10 across the screen in the Item Range Definition)? In placing RV in box 1 and XP in box 2, the RVs will be + and XPs will be -. But, if selecting ONLY XPs, the program doesn't care where the placement is. All expenses will come out as positive numbers. To reverse the sign on these items, list in the calculation field of the Column screen the Item Number minus the Item Number minus the Item Number (I05-I05-I05). This will take the value for that XP account and subtract it to zero and then subtract it again to get the negative total. In looking at the original coding of the program, one may see that this is exactly what the program is doing when it finds RV in box 1 and XP in box 2. The code doesn't do anything unless those two boxes are completed - you must do the calculation yourself.

5) TOTAL LINE PRINTING WITH EASE - In creating our reports and defining total lines, always keep the General Ledger structure in mind. While visiting screen #2 (Item Selection screen), be thinking of where your totals will appear, at what levels, and how much detail is to appear. If we would like to see totals for the different GL parts, select these items so we may sort by them in the screen that follows. It is far easier (and surer) to use the command BREAK(x) to total at the sort rather than use the AFTER command (this was discussed earlier). To insert sub-total lines or to follow the Fund Total or Account Total, use the same location for the sub-total line, BREAK(x). The sub-total will print following the original in the order you entered the Total Line screens. Look in the lower right-hand corner of the screen to find the record number for the sequence.

9 Appendix B: SACS Overview

9.1 SACS

SACS is the acronym for the "Standard Accounting Code Structure" for the State of California Department of Education. It is the accounting standards that California K-12 schools are required to use in all state reporting. In order to assist our clients in complying with SACS reporting requirements, SunGard Bi-Tech has developed utilities for extracting data from our system in a useful format, and a rule-based utility to ensure that data will not be entered that doesn't conform to the SACS requirement and that it cannot be entered into the system. Using the SACS utilities in IFAS requires creating common codes and other setup considerations.

9.2 IFAS has two general processes that relate to SACS

1. A process to take information from the IFAS GL and convert it into a format that can be loaded into the State's reporting software.

2. A process that takes the accounting rules imposed by the state and loads them into an IFAS rule database. These rules are then checked in real-time as users enter information into IFAS. If a user tries to use an accounting combination that is not allowed the software will block or show a warning.

9.3 SACS Allowable Combinations and IFAS

The state of California supplies a set of tables of allowable code combinations. SunGard Bi-Tech has written some utilities to allow you to take these files from the state’s web site and load them into the Rule Logic (GLUPRU) of IFAS. This will cause IFAS to validate transactions against these tables in real time so that no invalid combination will be entered.

SACS compliance requires careful consideration during the initial planning states of your accounting structure. If SACS compliance is required for your organization, refer to the California section of the State Regulatory Reporting guide.