Jenzabar CX

Requisitioning, Purchasing, and Accounts Payable

JENZABAR

Technical Manual

Filename: tmrpa
Distribution date: 01/05/2002

Contact us at www.jenzabar.com

JENZABAR, INC.
REQUISITIONING, PURCHASING, AND ACCOUNTS PAYABLE TECHNICAL MANUAL

TABLE OF CONTENTS

SECTION 1 - USING THIS MANUAL

Overview

Introduction

The Requisitioning, Purchasing and Accounts Payable products contain six applications:

- Requisition
- Approval
- Assign Buyer
- Purchase
- Accounts Payable
- Receiving

The Requisitioning product enables you to enter and approve requisition information about items your institution orders and pays for using the Purchasing and Accounts Payable product. You can also approve purchase orders and invoices, and track receiving information about the items ordered.

The Mass Invoice product as a stand-alone product used by some clients to enter invoice information that is not associated with requisitions or purchase orders. While this product is not associated with the Requisitioning, Purchasing and Accounts Payable products, it is documented in this technical manual.

Purpose of This Manual

This manual provides technical information required to install, customize, and maintain the Jenzabar Requisition, Purchasing and Accounts Payable products, and the Mass Invoice product.

Intended Audience

This guide is for use by those individuals responsible for the installation, customization, and maintenance of the CX.

How to Use This Manual

If you are not familiar with the processes and features of the Requisitioning, Purchasing, Assign Buyer, Accounts Payable, Mass Invoice and Receiving applications, read the manual for:

- Detailed reference information about how the applications works
- Procedures for customizing and maintaining the applications

If you are familiar with the processes and features of the Requisitioning, Purchasing, Assign Buyer, Accounts Payable, Mass Invoice and Receiving applications, and just need specific reference information, or a procedure, look through the Table of Contents or Index and refer to the pages you need.

Product Differences

This manual contains information for using all features developed for the Purchasing and Accounts Payable product. Your institution may or may not have all the features documented in this manual.

Structure of This Manual

This manual contains both general reference information and procedures for performing installation and maintenance on the Requisition, Purchasing and Accounts Payable products. The organization of the manual is as follows:

Related Documents and Help

The following resources are also available to assist you in implementing, supporting, and maintaining the Requisitioning, Purchasing and Accounts Payable products.

QuickMate online help
Using QuickMate
Getting Started User Guide

UNIX-based help
Help command (<Ctrl-w>) in screens and menus

User guides
Using Requisitioning
Using Purchasing and Accounts Payable
Getting Started User Guide

Technical manuals
CX Implementation and Maintenance Technical Manual
CX System Reference Technical Manual

Conventions Used in This Manual

Introduction

Jenzabar has established a set of conventions to help you use this manual. The list of conventions presented below is not exhaustive, but it includes the more frequently-used styles and terms.

Style Conventions

Jenzabar technical manuals observe the following style conventions.

Boldface type
Represents text that you type into the system (e.g., Type **UNDG**.) and command names or keys you use to execute a command or function in a procedure (e.g., **Finish**, **<Enter>**).

Bulleted lists
Show items not ranked or without a sequential performance.

CAUTION:
Indicates a caution or warning of a potential risk or condition.

<Enter>
Represents the Enter, Return, Line Feed, or ↵ key on your keyboard.

Italic type
Is used in any of these ways:

- To represent a new or key term
- To add emphasis to a word
- To cross-reference a section of text
- To represent a variable for which you substitute another variable (e.g., substitute *filename* with an appropriate filename)

<Key name>
Represents a key that you must press.

Note:
Indicates a note, tip, hint, or additional information.

Numbered lists
Show ranking of items or sequence of performance.

Parentheses
When used around a field name, indicate the field is unlabeled. The field description includes the location of the field.

Quotation marks
Represent information written in this guide exactly as it appears on the screen (e.g., The message, "Now Running..." appears.).

Jenzabar-Specific Terms

Some terms used in this manual may be unfamiliar to you, either because they are terms you have not used before or because Jenzabar has assigned a slightly different meaning to a familiar term. The following list identifies and explains the most common Jenzabar-specific terms:

Application
A group of one or more software programs that enables you to perform a particular procedure, such as entering student information.

Data

Specific information you enter into fields on a particular data entry screen.

Enter

To type information on a keyboard and execute by either of the following actions:

- Pressing the **<Enter>** key
- Clicking on the **OK** button
- Selecting **Finish**

F key

Any of the function keys located on your keyboard (e.g., **<F1>**).

Hot key

The capitalized and underlined (or highlighted) letter of a command on a menu.

ID

The number assigned to each student or organization associated with your institution (e.g., 12345).

Institution

An established organization of postsecondary education that supports all operating functions (e.g., a college or university).

Parameter

A variable in the system that is given a constant value for a specific application (e.g., a date can be a parameter for producing a report).

Select

To execute a command by any of the following actions:

- Performing the keystrokes
- Pressing the hot key
- Highlighting the command or option and pressing the **<Enter>** key
- Clicking with the mouse

System

The Jenzabar suite of products; the CX.

Keystrokes

When you see two keys separated by a dash (e.g., **<Ctrl-c>**), hold down the first key (**Ctrl**) while pressing the second (**c**).

SECTION 2 - REQUISITIONING PROCESSES

Overview

Introduction

This section contains a brief description of the purpose of the applications included in the Requisitioning product. A process flow diagram and description provide more specific information about how the requisition and approval processes work.

Purpose of the Module

The Requisitioning product provides the capability of creating and submitting requisitions for items to be purchased, and to automatically process the approval of those requisitions before they are available for further processing in Purchasing.

Applications

The Requisitioning product consists of the following two applications:

Requisition

Enables you to gather information about specific items being requisitioned for purchase. The types of information this application tracks include:

- Name of requisitioner
- Required delivery date
- Quantity of item
- Item code and description
- Estimated cost, including discounts and freight charges
- Account number(s) to be charged
- Suggested vendor name and address

Approval

Enables you to approve or deny the purchase of requisitioned items, and automatically routes the approval notice to the appropriate approver. This process can use either a vertical or horizontal method (or a combination of the two) for performing approvals.

Note: *Vertical processing* means one person must provide approval for a requisition, purchase order or invoice before a second person can approve it.

Horizontal processing means that several people can approve requisitions, purchase orders or invoices simultaneously. *All* must approve it before the document moves to the next approval step or on to the next processing step (e.g., Purchasing or Accounts Payable).

Background Knowledge

The following describes the necessary background information you should know to implement and support the Requisitioning product.

UNIX

Know the following about the UNIX operating system:

- Csh environment and commands
- Editor commands (e.g., vi)

INFORMIX-SQL

Know the following about the following INFORMIX tools:

- SQL database
- PERFORM screens
- ACE reports

Jenzabar CX database tools and utilities

Know how to use the following database tools:

- MAKE processor
- Schemas
- Program screens
- The SMO process

Jenzabar CX

Know the following about the Jenzabar CX standard product:

- CX directory structure
- The menu processor
- The CX database engine

QuickMate features

Know the following about the Jenzabar Graphical Server:

- Client/Server processing
- Network settings
- Keyboard settings
- Mouse settings
- GUI mode commands

C Programming

If you want to modify any CX programs to meet unique needs at your institution, you must know how to use the C programming language and have an in-depth knowledge of the CX code.

Requisitioning policies and procedures

Know answers to the following questions:

- What are your institution's procedures for approving requisitions and purchase orders?
- How does your institution plan to implement the requisition and approval processes (e.g., department by department)?
- How many stations do you want to set up for requisitions, to provide for document numbering sequences?
- What is the institution's policy for approving requisitions or purchase orders for which the budget is not sufficient?

Requisition Process Description

You access the *requisition* application by selecting a menu option that defines the type of requisition you want to generate. The menu options you can select from are:

- Requisition for Purchase
- Requisition from Invoice
- Requisition for Check

When a user submits a requisition, *requisition* submit logic verifies the following requirements:

- The requisition must include at least one item
- All required fields must contain valid information
- The requisition has not previously been submitted

In addition, the program also checks the general ledger account(s) charged for the requisitioned item(s) to determine if sufficient funds exist in the budget to pay for the item(s). If the budget is insufficient, the program displays a warning message to the person submitting the requisition.

After verifying the requirements for a submitted requisition, the submit logic does the following:

- Adds records to the approval tables that maintain information for *requisition* and *approval*
- Notifies the appropriate approvers, via e-mail, that they have a requisition to approve

Approval Process Description

The approval process begins when a requisition, purchase order or invoice is submitted. In the first phase of the approval process, the *approval* program goes through the following steps:

1. Determines who must approve the document, based on approval type criteria (e.g., ID, account or commodity code).
2. Sorts approvers by authorization limit.
3. Notifies the first approver (vertical) or approvers (horizontal) by mail that a document needs approval.
4. After each approval (vertical) or group of approvals (horizontal) is received:
    - Notifies the submitter that the document has been approved by the approver.
    - Determines who must approve the document next, then sends that approver mail to notify them that a document awaits his/her approval.
    - If the document is denied, notifies the submitter of the denial, including the reason.
5. The *approval* program continues this process until all approvals have been received. Then, the program notifies the person who submitted the document that it has been approved, and makes a requisition available to purchasing. If the document is a purchase order or an invoice, the program makes the document available to accounts payable for appropriate processing.

Module Relationships

Related Jenzabar CX Modules

The Requisitioning product interacts with several other applications in CX. The following list describes the relationships.

Purchase
Combines the requisition items to generate a purchase order from the requisition data.

Accounts Payable
Matches purchase order items to those received and matches an invoice to the listed items, before scheduling payment. Optionally, a Fixed Asset record can be created from the information gathered.

Receiving
Verifies that items ordered have been received in their proper condition.

Fixed Assets
Processes those items that are identified as fixed assets and generates a skeletal Fixed Asset record when the invoice is submitted for payment.

Vendor Entry
Provides direct access to permit users to enter ID information for vendors for whom no ID record exists on the database. It is also used to update current vendor information.

Budget Review
Provides access to information about account balances, including budget, actual and encumbered.

Assign Buyer
Enables users to assign requisitions to specific buyers for the purchasing function.

SECTION 3 - PURCHASING AND ACCOUNTS PAYABLE PROCESSES

Overview

Introduction

This section provides information on the purpose and process flow of the Purchasing and Accounts Payable product.

Purpose of Product

The Purchasing and Accounts Payable product provides the following capabilities:

- Assigning approved requisitions to buyers
- Creating purchase orders
- Creating invoices
- Tracking the receipt of goods
- Creating skeletal Fixed Asset records

Specifically, this product includes the following applications:

Accounts Payable

Enables you to create and submit invoices for payment of vendors as follows:

- For an associated purchase order
- Without an associated purchase order (direct or manual invoices)
- Without detailed line item information (simplified or fast invoices)
- For a requisition for a check

Assign Buyer

Enables you to assign and reassign approved requisitions to a Purchasing buyer.

Mass Invoice

Enables you to enter manual invoices in batch, without invoice item detail.

Purchasing

Enables you to create and submit the following types of purchase orders:

- Blanket Purchase Order
- Invoice Type Purchase Order
- Open Purchase Order
- Standard Purchase Order

Receiving

Enables you to receive, return and track items:

- Against purchase order line items
- Against a purchase order
- Without a purchase order

Background Knowledge

The following list describes the necessary background information you should know to implement and support the Purchasing and Accounts Payable product.

UNIX

Know the following about the UNIX operating system:

- Csh environment and commands
- Editor commands (e.g., vi)

INFORMIX-SQL

Know about the following INFORMIX tools:

- SQL database
- PERFORM screens
- ACE reports
- ESQL

Jenzabar CX database tools and utilities

Know how to use the following database tools:

- MAKE processor
- Schemas
- Macros
- Includes
- Program screens
- The SMO process

Jenzabar CX

Know the following about the Jenzabar CX standard product:

- CX directory structure
- The menu processor
- The CX database engine

QuickMate features

Know the following about the CX Graphical Server:

- Client/Server processing
- Network settings
- Keyboard settings
- Mouse settings
- GUI mode commands

C Programming

If you want to modify any CX programs to meet unique needs at your institution, you must attend the Jenzabar Platform class and know how to use the C programming language.

Purchasing and Accounts Payable policies and procedures

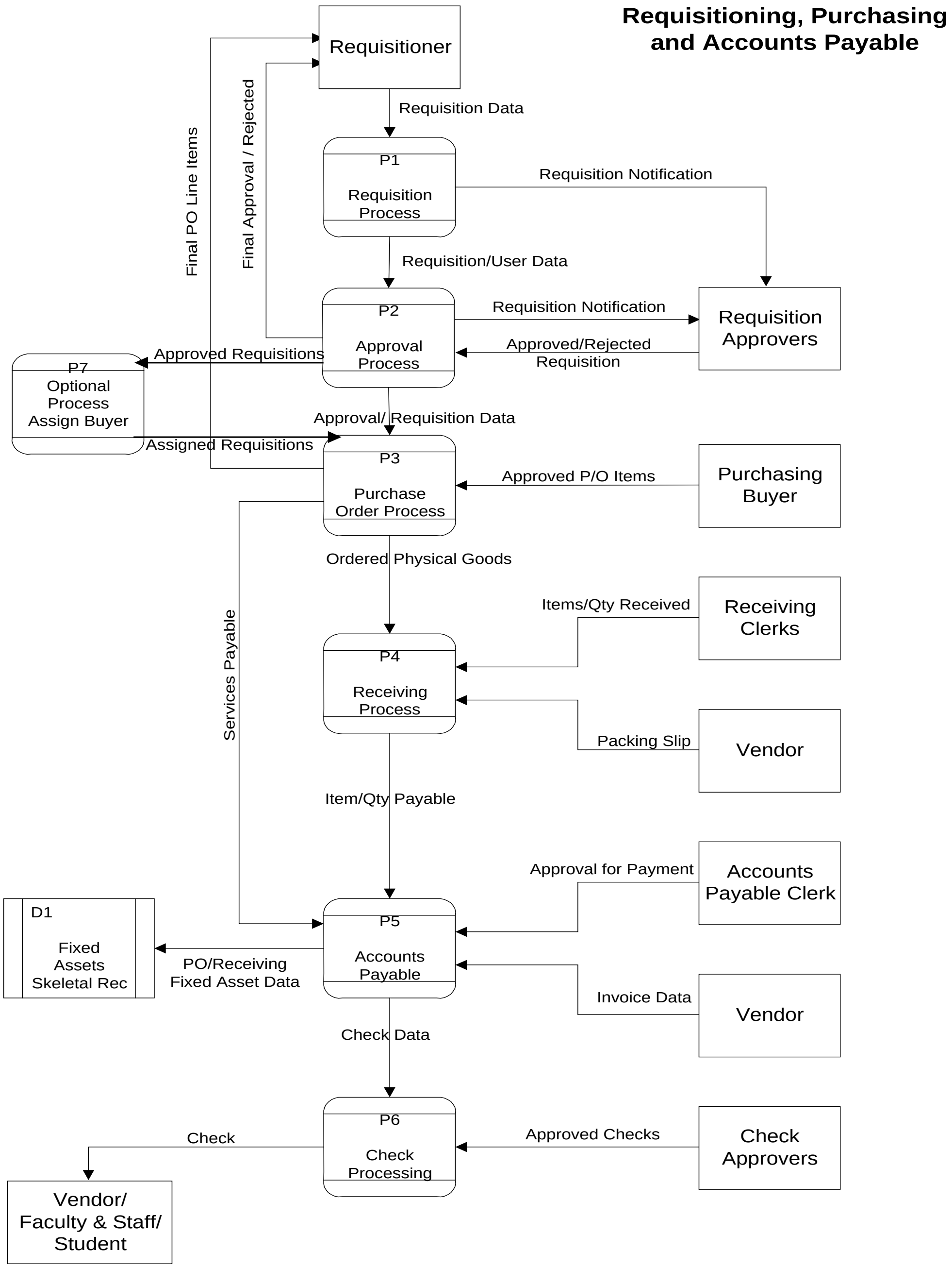
Know answers to the following questions:

- What are your institution's accounting policies?
- What procedures does your institution use to create entries for accounts payable?
- Does your institution want to assign approved requisitions to specific buyers?

Process Flow

Diagram

The following diagram shows the process flow of the Purchasing and Accounts Payable product. For more information about program interrelationships and detailed data flow diagrams, see the sections in this manual that describe the specific programs.



Process Description

The following list describes the processes involved with the Purchasing and Accounts Payable product.

1. The Assign Buyer application accesses approved requisitions, and assigns buyers for the requisitioned items.
   Note: This process is optional.
2. The buyer prepares purchase orders in the Purchasing application, then orders the goods or services.
3. The vendor or supplier provides the goods or services. A packing slip accompanies received goods.
4. The receiver checks the received goods against the packing slip, and records the receipt in the Receiving application.
5. The vendor or supplier submits an invoice for the goods or services.
6. The Accounts Payable clerk enters the invoice, using Invoice Entry.
   Note: If the institution’s policies do not require the tracking of requisitions or purchase orders for every invoice, then users can also enter invoices using the Mass Invoice application. The two types of invoices (from Invoice Entry and from Mass Invoice) are maintained separately; you can query invoices only from within the application in which they were created. For example, if users create an invoice in Mass Invoice, they can view the invoice only from Mass Invoice.
7. The Accounts Payable clerk matches the requisition, purchase order, packing slip and invoice, and uses the Accounts Payable application to prepare a check to pay the invoice.

Relationships Among Programs, Forms and Processes

Introduction

A close relationship exists between the programs in the Requisitioning product (*requisition* and *approval*) and the programs in the Purchasing and Accounts Payable product (*acctspay*, *assign*, *purchase* and *receive)*. The purpose of this section is to explain the relationships between the programs, and to show how CX uses forms to reflect the relationships.

Diagram

The following diagram shows the relationship between the programs and the forms used.

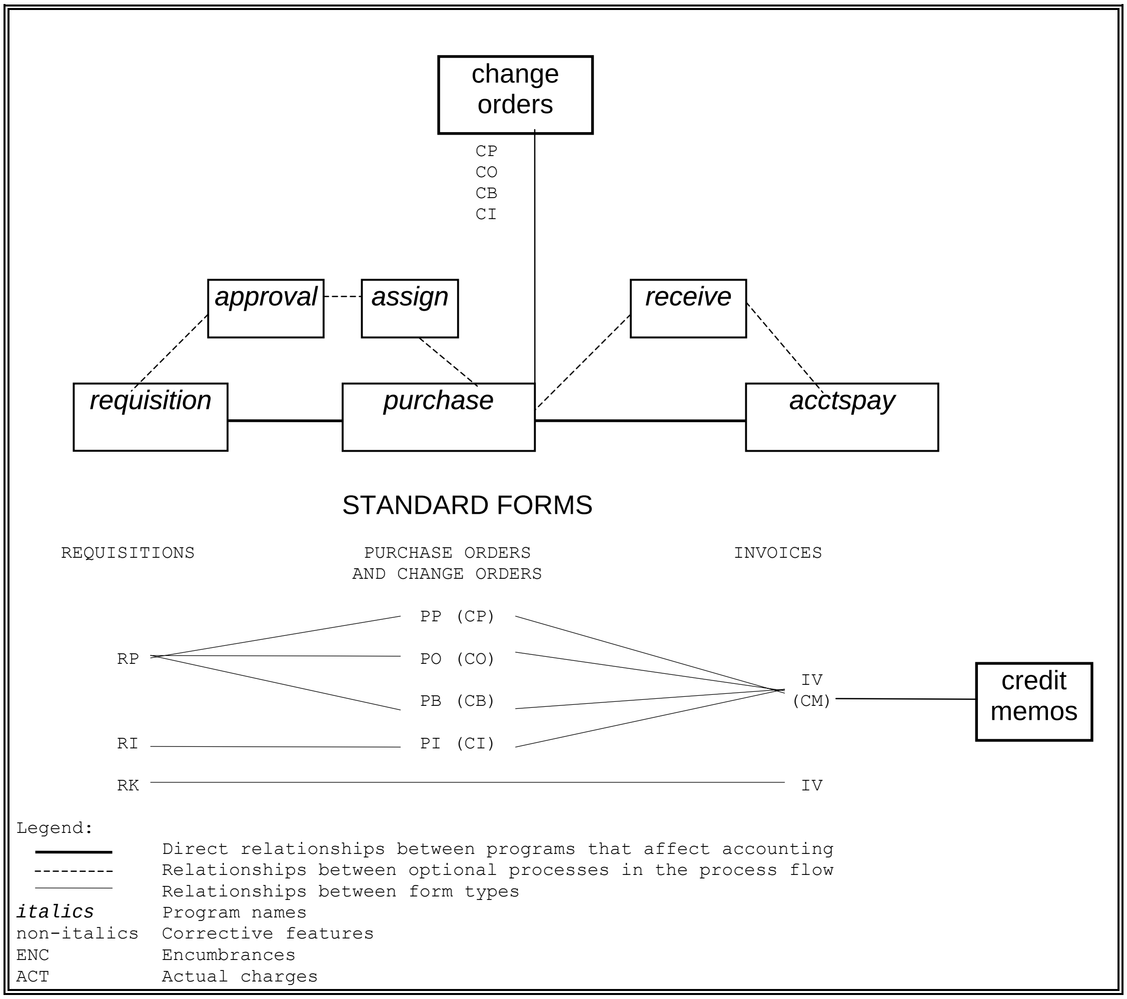


Diagram Description

The following describes the diagram on the previous page. If required, all the standard form type names (e.g., RP, CP, PB) can be modified during implementation; however, Jenzabar recommends that institutions modify these values only if they conflict with other codes in the Document table. Changes to standard form type names require changes to the standard table values. For more information about changing the form type names and the related table values, see *Purchasing and Accounts Payable Macros and Includes* and *Installing the Purchasing and Accounts Payable Processes* in this manual.

1. The process begins, for most procurement transactions, in the *requisition* program, when a user formally requests the purchase of goods or services. The request can impact CX in any of three ways, designated by the form type used:
   - A requisition for a purchase order (i.e., a routine request, when the purchase order is completed before the goods or services are obtained). This requisition type uses the RP form type.
   - A requisition from an invoice (i.e., a reimbursement for goods or services already obtained and for which the receipt becomes the invoice). This requisition type uses the RI form type.
   - A requisition for a check (i.e., a request for a check without any supporting invoice, as in the case of an advance). This requisition type uses the RK form type.
2. Requisitions are submitted to the *approval* process, and optionally routed to the *assign* process. Neither of these processes impacts the accounting for procurement transactions.
3. The *purchase* process creates purchase orders. In CX, the creation of the purchase order causes the actual encumbrance of the dollars to pay for the goods and services. Purchase orders can be of the following types:
   - PP (one-time purchase orders, e.g., for non-recurring purchases)
   - PO (open purchase orders, e.g., for repetitive purchases that do not have a definite limit, such as a utility bill)
   - PB (blanket purchase orders, e.g., for repetitive purchases that have a definite limit, such as an equipment or textbook budget)
   - PI (purchase orders that relate to items already purchased, where the receipt serves as the invoice)

   The first three purchase order types (PP, PO, and PB) can only include items from the RP requisition type. The PI purchase order type can only include items from the RI requisition type.

   If changes to the ordered goods or services occur during the *purchase* process, the program uses change orders to track the changes. The program creates a new Change Order record (chgorder_rec) each time the order changes. The form type for the change order (CP, CO, CB or CI) depends on the type of purchase order that it changes (PP, PO, PB or PI respectively).
   Example: Assume that an individual orders 5 units of a product through a one-time purchase order type of PP. During the *purchase* process, the individual notifies Purchasing that he needs 10 units, instead of 5. Before Purchasing orders the merchandise, the individual contacts Purchasing again, stating that he needs 9 units, not 10.
   In this situation, *purchase* maintains the original PP for 5 units, a CP change order to reflect the need for 10 units, and an additional CP change order for the 9 units actually ordered. The Purchase Order Header record (pohead_rec) exists for all three orders, providing a complete audit trail. To simplify the querying process, queries access only the most recent change order.

4. When the goods arrive, the receive program tracks the quantity and condition of the merchandise. *Receive* does not impact the accounting of procurement transactions, but enables institutions to store data about their inventory and received goods, and to provide information to Accounts Payable. Generally, the *receive* program does not track the completion of services.

5. The *acctspay* program uses information from *purchase* and *requisition* to create actual charges against accounts, and to unencumber amounts that were encumbered in the *purchase* process. Form types in *acctspay* are IV. An invoice is required to initiate an accounts payable transaction.

Product Relationships

Related Jenzabar Products

The Purchasing and Accounts Payable product interacts with several other Jenzabar products. The following list describes the interrelationships.

Requisitioning
Provides automated requisitioning for goods, services and reimbursements, providing information for purchase orders and invoices.

Fixed Assets
Accepts skeletal Fixed Asset records created in Accounts Payable to track acquisitions of long-lived assets.

Check Processing
Uses submitted invoices from Accounts Payable to process and print checks.

General Ledger
Provides budget review information. Also uses submitted purchase orders to create encumbrances, and submitted invoices to create actual charges and relieve encumbrances.

SECTION 4 - REQUISITIONING TABLES AND RECORDS

Overview

Introduction

This section provides you with reference information about each table and record associated with the Requisitioning product. Schema files for these and other related tables and records are identified by their UNIX filename, INFORMIX filename, and table/record name.

The following tables are described in this section:

- Account Authorization table (auth_acct_table)
- Building table (bldg_table)
- Commodity Code Authorization table (auth_comm_table)
- Commodity Code table (commc_table)
- Document table (doc_table)
- Denial table (denial_table)
- Facility table (facility_table)
- Form Type table (form_table)
- ID Authorization table (auth_id_table)
- Requisition Units table (units_table)
- User Identification table (userid_table)

The following records are described in this section:

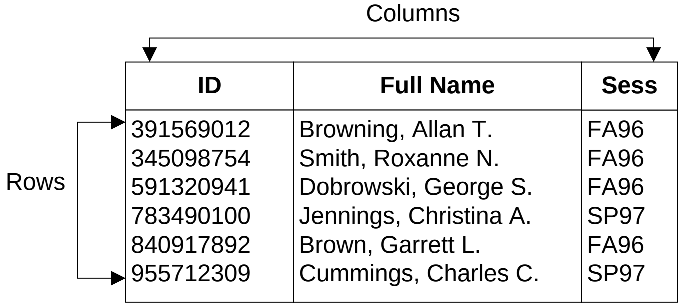
- Approval record (approv_rec)
- Approved Requisition Line Item Entries record (appritem_rec)
- Denial Comment record (denial_blob)
- Identification record (id_rec)
- Pending Requisition record (pendreq_rec)
- Requisition Address Entries record (reqaddr_rec)
- Requisition Denial record (denial_rec)
- Requisition Header Comment record (reqhead_blob)
- Requisition Header Entries record (reqhead_rec)
- Requisition Line Item Account Entries record (reqacct_rec)
- Requisition Line Item Comment record (reqline_blob)
- Requisition Line Item Entries record (reqline_rec)

Alphabetical Organization

The tables and records appear in alphabetical order in this section.

What Is an SQL Table?

In a relational SQL database, a table is an organized set of any kind of data, regardless of its purpose for validation or information maintenance. The basic unit of organization of a table is a column, a category of data. A table can have multiple columns, and columns typically contain multiple rows of data.



ID	Full Name	Sess
391569012	Browning, Allan T.	FA96
345098754	Smith, Roxanne N.	FA96
591320941	Dobrowski, George S.	FA96
783490100	Jennings, Christina A.	SP97
840917892	Brown, Garrett L.	FA96
955712309	Cummings, Charles C.	SP97

What Is a Jenzabar CX Table?

CX makes name distinctions in the usage of database tables. A *table* in CX contains information that remains static and is denoted with the *_table* extension. For example, the State table, named *st_table,* contains the list of states in the United States of America. On the CX menu, you can access most tables in *Table Maintenance* menus.

What Is a Jenzabar CX Record?

CX makes name distinctions in the usage of database tables. A *record* in CX is a table that contains information that changes on a regular basis and is denoted with the *_rec* extension. For example, the Alternate Address record, named aa*_rec*, contains any other address where the individual can be contacted, such as a summer or business address. You access records via CX program screens, scroll screens, and PERFORM screens.

Common Tables and Records

Programs in the Requisitioning product use some tables and records that are used throughout CX. The following common tables and records are used:

- Building table (bldg_table)
- Document table (doc_table)
- Facility table (facil_table)
- Identification (ID) record (id_rec)
- User Identification (Userid) table (userid_table)

General Ledger Tables and Records

Programs in the Requisitioning product use several tables and records that are used in the General Ledger product in CX. Complete descriptions and setup instructions for the General Ledger tables appear in *General Ledger Technical Manual*.

Shared Tables and Records

The following lists the shared tables and records in the Requisitioning product, and other products and programs that use them.

Note: If you change the following table and record files, you must reinstall each associated product.

Building table (bldg_table)

The student management product

Document table (doc_table)

Facility table (facil_table)
All student management and some financial products, including the following programs:
Assign, Billing, Course Entry, Libusr, Purchase, Receive, Requisition

Identification record (id_rec)
All products

User Identification table (userid_table)
All products, including the following programs:
Libcars, Purchase, Requisition

Required Tables and Records

The following records are required to run the features of the Requisitioning product.

Account Authorization table (auth_acct_table)

Entries for approvals based on the accounts charged for requested items (e.g., approver X approves all requisitions charged to the account 10-1003-6000)

Commodity Code Authorization table (auth_comm_table)
Entries for approvals based on the type of commodity requested (e.g., approver X approves all requisitions for computer equipment)

Commodity Code table (commc_table)
Entries for the commodity codes that requesters use on requisitions, purchase orders, invoices

Denial table (denial_table)
Entries for standard reasons used for denying requisitions

Document table (doc_table)
Entries for each record in the Form table

Form Type table (form_table)
Entries for each different document type (e.g., IV for invoices)

ID Authorization table (auth_id_table)
Entries for approvals based on the IDs of users (e.g., approver X approves all requisitions for the user whose ID is 12345)

Identification record (id_rec)
Records for each user and/or vendor

Requisition Units table (units_table)
Entries for each type of unit purchased (e.g., each, gallon)

User Identification table (userid_table)
Entries for each user

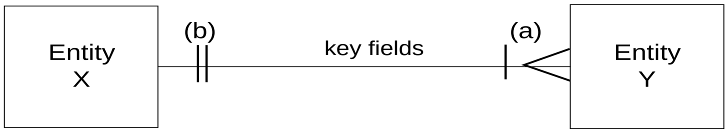
Entity Relationship Diagram

Key to Entity Relationship Diagrams

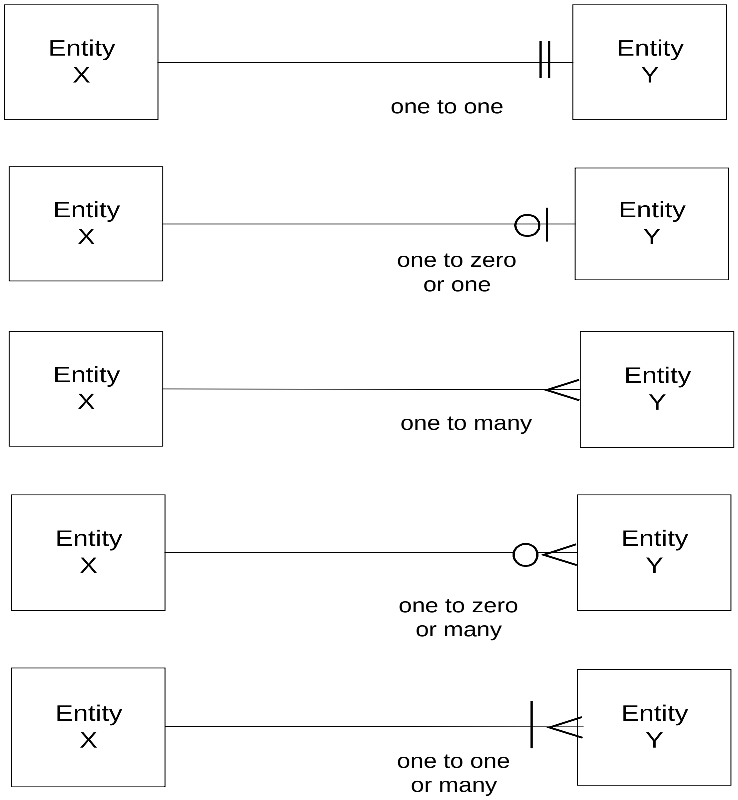
The diagrams on the following pages show the relationship between the tables and records used by the applications in the Requisitioning product. This diagram provides a key to interpreting the entity relationships in these diagrams.

Interpret the diagrams as follows:

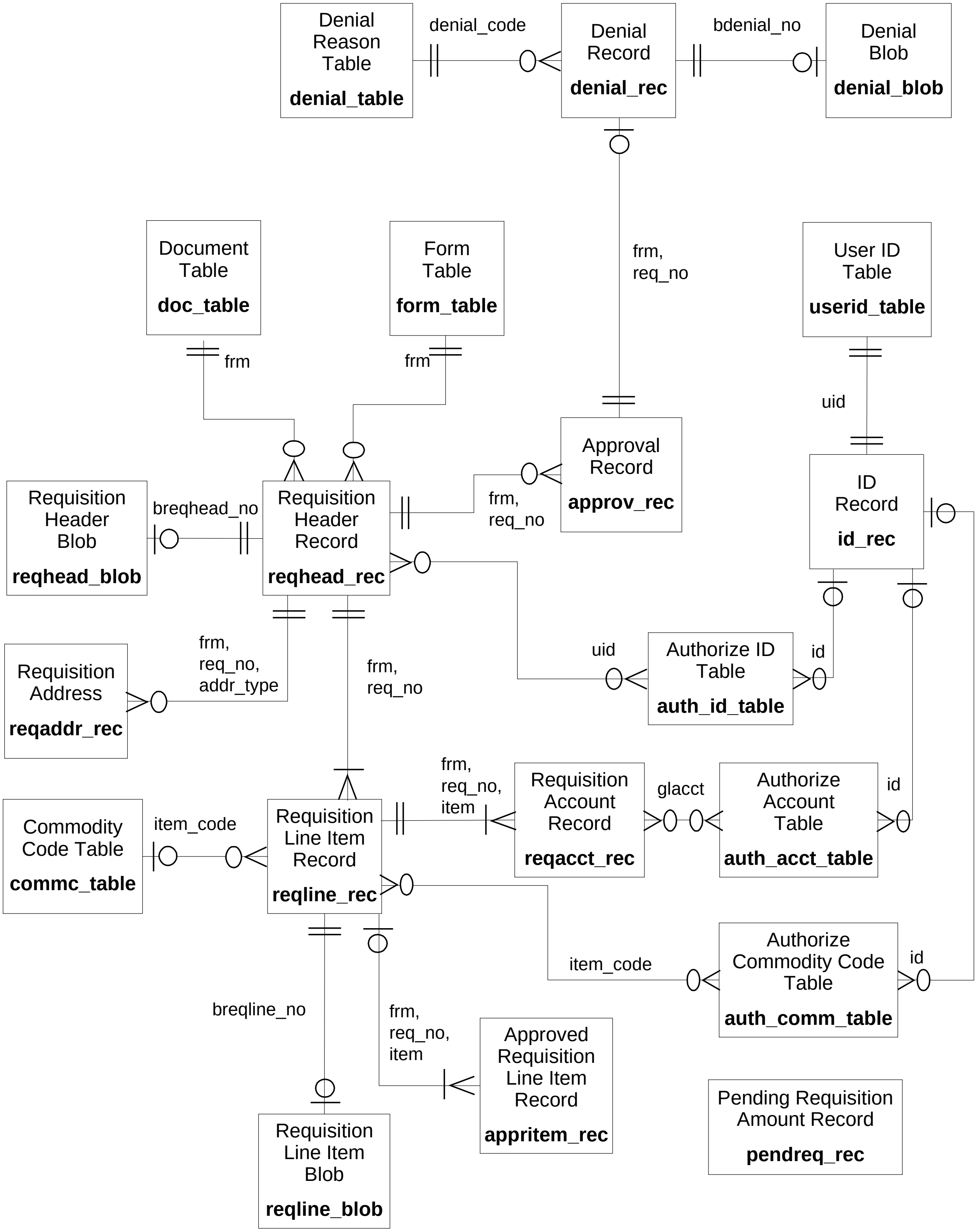
- Entity X has a one to (a) one or many relationship to Entity Y.
- Entity Y has a one to (b) one relationship to Entity X.



Relationship symbols and their meanings:



Requisition and Approval Applications - Entity Relationship Diagram



Requisitioning Schemas

Introduction

Schema files define database files and associated fields in the Jenzabar data dictionary. You can access schema files associated with the Requisitioning product in the following directory paths:

- $CARSPATH/schema/financial
- $CARSPATH/schema/common

File Naming Conventions

CX makes distinctions in the naming of schema. For schema files containing definitions of CX tables, the UNIX file name begins with the letter *t* followed by characters describing the table's English name (e.g., *tst* for the State table). For schema files containing definitions of CX records, the UNIX file name describes the record's English name (e.g., id for ID record).

The first line in a schema file, after revision information, specifies the INFORMIX database table or record that the schema defines. For example, *st_table* (State table) is specified in the *tst* schema file.

Field Descriptions

Schema files contain descriptions of each field defined in a table or record. You can view descriptions of the fields in Requisitioning tables and records by accessing the schema files.

Schema File Reports

Standard CX includes three reports that provide information about the contents of the schema files. When table implementation begins, you can run the reports to provide the installation team with information about the tables and their fields.

Select the report options from the following menu path:
System Management: Data Dictionary menu

The reports are as follows:

dbefield
Lists each column in the database by table, including its name, short and long descriptions, field type (e.g., char or date) and size.

dbefile
Lists the tables that relate to each track area (e.g., ADM or COM), including the table name, description and purpose.

dbetrack
Combines the contents of dbefield and dbefile, displaying the tables for each product area and the columns in each table.

Requisitioning Tables and Records

Introduction

The following listing includes the primary tables and records that are used in the Requisitioning product. This listing indicates each table/record's purpose and association with programs and other tables and records. The tables and records appear in the following order:

- Common tables and records
- General Ledger tables and records
- *Approval* tables and records
- *Requisition* tables and records

Note: In this listing, the listed programs are within the Requisitioning product, and the listed products and applications are outside the Requisitioning product.

Common Schema Files

The following table lists the common schema files that define the fields in tables and records used by the Requisitioning product.

Building table

UNIX filename: tbldg

Informix filename: bldg_table

Schema location: $CARSPATH/schema/common

Purpose: Contains descriptions of known buildings and defines an identifying code for each.

Program interrelationships: *approval*, *requisition*

Product interrelationships: Events Management, Registration

Table/record interrelationships: reqhead_rec

Commodity Code table

UNIX filename: tcommc

Informix filename: commc_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the Commodity Code record associated with the Item Code and contains a Central Store Availability flag.

Program interrelationships: *requisition, approval*

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: reqline_rec

Document table

UNIX filename: tdoc

Informix filename: doc_table

Schema location: $CARSPATH/schema/common

Purpose: Defines the types of source documents used by the system and the code that represents each document.

Program interrelationships: requisition

Product interrelationships: General Ledger, Cashier, Lib, Development, Financial Aid, Payroll, Purchasing and Accounts Payable

Table/record interrelationships: form_table, invoice_rec, reqhead_rec, vch_rec

Facility table

UNIX filename: tfacil

Informix filename: facility_table

Schema location: $CARSPATH/schema/common

Purpose: Describes the room facilities available and defines the campus and building where those rooms are located.

Program interrelationships: requisition

Product interrelationships: Purchasing and Accounts Payable, Registration, Student Affairs and Housing

Table/record interrelationships: reqhead_rec

Form Type table

UNIX filename: tform

Informix filename: form_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the various form types for the Requisition, Purchasing and Receiving applications.

Program interrelationships: requisition

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: doc_table, reqhead_rec

General Ledger Substitution table

UNIX filename: tglsub

Informix filename: glsub_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the segments of the General Ledger account number that trigger the General Ledger Account Auto-Fill feature.

Program interrelationships: Purchasing, Accounts Payable, Mass Invoice, Recurrent, General Ledger, Approval

Product interrelationships: General Ledger, Purchasing and Accounts Payable

Identification record

UNIX filename: id

Informix filename: id_rec

Schema location: $CARSPATH/schema/common

Purpose: Contains the names and general information for all individuals and entities.

Program interrelationships: approval, *requisition*

Product interrelationships: Used by all products

Table/record interrelationships: aa_rec, auth_acct_table, auth_comm_table,

Units table

UNIX filename: tunits

Informix filename: units_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines units of an item quantity in a requisition (e.g. 2 crates, 5 kits, 1.5 lbs)

Program interrelationships: requisition

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: reqline_rec

User Identification table

UNIX filename: tuserid

Informix filename: userid_table

Schema location: $CARSPATH/schema/common

Purpose: Provides a link between the system (login) user ID number and the Jenzabar database ID number.

Program interrelationships: approval, *requisition*

Product interrelationships: Used by all products

Table/record interrelationships: id_rec

General Ledger Tables and Records

The following General Ledger tables and records relate to the Requisitioning product.

- General Ledger Account record (gla_rec)
- General Ledger Amount record (gl_amt_rec)
- General Ledger Entries record (gle_rec)
- Journal record (vch_rec)
- Subsidiary Account record (suba_rec)
- Subsidiary Balance record (subb_rec)
- Subsidiary Entry record (sube_rec)
- Subsidiary Transaction record (subtr_rec)
- Subsidiary table (subs_table)
- Subsidiary Total record (subt_rec)
- Transactions record (gltr_rec)

For more information about these tables and records, see *General Ledger Technical Manual.*

Approval Tables and Records

The following tables and records relate to the *approval* program.

Account Authorization table

UNIX filename: taprvact

Informix filename: auth_acct_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines who is authorized to approve requisitions based on accounts for the specified dollar limits.

Program interrelationships: approval, *requisition*

Product interrelationships: none

Table/record interrelationships: id_rec, reqhead_rec, approve_rec

Approval record

UNIX filename: approv

Informix filename: approv_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines those people who need to approve a requisition, along with the status of the approvals.

Program interrelationships: approval, *requisition*

Product interrelationships: none

Table/record interrelationships: appritem_rec, auth_acct_table, auth_comm_table, auth_id_table, denial_table

Approved Requisition Line Item Entries record

UNIX filename: appritem

Informix filename: appritem_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines approved requisition line items.

Program interrelationships: approval, *requisition*

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: poacct_rec, poline_rec, reqacct_rec, reqline_rec

Commodity Code Authorization table

UNIX filename: taprvcom

Informix filename: auth_comm_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines who is authorized to approve requisitions based on the commodity codes contained on the requisitions.

Program interrelationships: *approval*, *requisition*

Product interrelationships: none

Table/record interrelationships: id_rec, reqhead_rec, approve_rec

Denial Comment record

UNIX filename: bdenial

Informix filename: denial_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the denial reasons on a requisition, along with any special instructions.

Program interrelationships: *approval*, *requisition*

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: denial_rec

Denial table

UNIX filename: tdenial

Informix filename: denial_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the valid denial codes for the Approval process.

Program interrelationships: *approval*, *requisition*

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: denial_rec

ID Authorization table

UNIX filename: taprvid

Informix filename: auth_id_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines who is authorized to approve requisitions for individuals at specified dollar limits.

Program interrelationships: *approval*, *requisition*

Product interrelationships: none

Table/record interrelationships: id_rec, reqhead_rec, approv_rec

Requisition Denial record

UNIX filename: denial

Informix filename: denial_rec

Schema location: $CARSPATH/schema/financial

Purpose: Tracks the denial information when a requisition is denied.

Program interrelationships: requisition, approval

Product interrelationships: none

Table/record interrelationships: denial_blob, reqhead_rec

Requisition Tables and Records

The following tables and records relate to the Requisition program.

Pending Requisition record

UNIX filename: pendreq

Informix filename: pendreq_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains pending requisition amounts for all requisitions from submission until placement on purchase orders (i.e., submitted but not yet approved and ordered requisitioned items). Pending requisitions can encumber budgeted dollars if desired. The system maintains this record.

Program interrelationships: requisition, approval

Product interrelationships: Purchasing and Accounts Payable, Budget Review

Table/record interrelationships: none

Requisition Address Entries record

UNIX filename: reqaddr

Informix filename: reqaddr_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines requisition address data to reduce address storage space.

Program interrelationships: requisition, approval

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: reqhead_rec

Requisition Header Comment record

UNIX filename: breqhead

Informix filename: reqhead_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the Requisition Header.

Program interrelationships: requisition, approval

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: reqhead_rec

Requisition Header Entries record

UNIX filename: reqhead

Informix filename: reqhead_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines Requisition Header information.

Program interrelationships: requisition, approval

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: appritem_rec, doc_table, form_table, reqaddr_rec, reqhead_blob, reqline_rec

Requisition Line Item Account Entries record

UNIX filename: reqacct

Informix filename: reqacct_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines requisition line item account information.

Program interrelationships: requisition, approval

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: reqline_rec

Requisition Line Item Comment record

UNIX filename: breqline

Informix filename: reqline_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the requisition line item.

Program interrelationships: approval, *requisition*

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: reqline_rec

Requisition Line Item Entries record

UNIX filename: reqline

Informix filename: reqline_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines requisition line item information.

Program interrelationships: approval, requisition

Product interrelationships: Purchasing and Accounts Payable

Table/record interrelationships: appritem_rec, commc_table, reqacct_rec, reqhead_rec

SECTION 5 - PURCHASING AND ACCOUNTS PAYABLE TABLES AND RECORDS

Overview

Introduction

This section provides reference information about each table and record associated with the Purchasing and Accounts Payable product. The following tables and records are specific to the applications with this product:

Accounts Payable

- Check Group record (ckgrp_rec)
- Check Request record (ckreq_rec)
- Invoice Account Entries record (invacct_rec)
- Invoice Header Comment record (invhead_blob)
- Invoice Header Entries record (invhead_rec)
- Invoice Line Item Comment record (invline_blob)
- Invoice Line Item Entries record (invline_rec)
- Payment Terms table (pay_term_table)

Assign Buyer

- Buyer table (buyer_table)

Mass Invoice

- Invoice Category table (invctgry_table)
- Invoice Charge record (invchg_rec)
- Invoice record (invoice_rec)
- Invoice Subsidiary record (invsubs_rec)

Purchasing

- Purchase Order - Change Order Entries record (chgorder_rec)
- Purchase Order Header Comment record (pohead_blob)
- Purchase Order Header Entry record (pohead_rec)
- Purchase Order Line Item Account Entries record (poacct_rec)
- Purchase Order Line Item Comment record (poline_blob)
- Purchase Order Line Item Entries record (poline_rec)
- Requisition Line Item Denial Entries record (denial_line_rec)
- Requisition Sort Fields table (sortreqfld_table)
- Requisition Sort Order table (sortreq_table)
- Vendor record (vnd_rec)

Receiving

- Line Item Return Entries record (return_rec)
- Method of Shipping Returned Goods table (shipmethod_table)
- Reason for Returning Goods table (retreason_table)
- Receiving Header Comment record (rechead_blob)
- Receiving Header Entries record (rechead_rec)
- Receiving Line Item Comment record (recline_blob)
- Receiving Line Item Condition table (cond_table)
- Receiving Line Item Entries record (recline_rec)
- Receiving Line Item Information record (recinfo_rec)
- Receiving Return Comment record (return_blob)

These and other applicable tables and records are listed with additional information under the headings *Purchasing and Accounts Payable Schemas* and *Purchasing and Accounts Payable Tables and Records* in this section.

Notes:

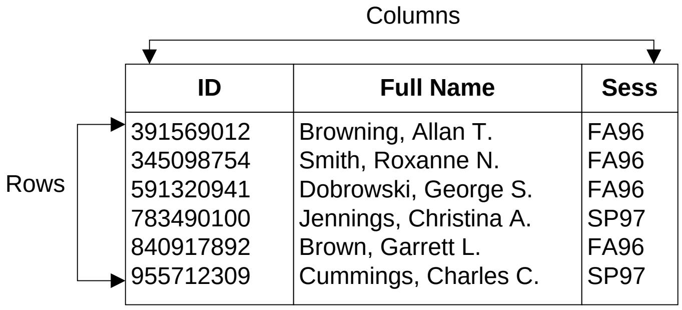
- In addition, this section includes the following tables used in the Vendor Entry program.
    - ˜ Vendor Quality table (venqual_table)
    - ˜ Vendor Type table (ventype_table)
- Vendor Entry uses the standard CX Entry library. For more information about the Entry library, see the *Technical Manual*.

Alphabetical Organization

The tables and records in this section appear in alphabetical order.

What Is an SQL Table?

In a relational SQL database, a table is an organized set of any kind of data, regardless of its purpose for validation or information maintenance. The basic unit of organization of a table is a column, that is, a category of data. A table can have multiple columns, and multiple rows of data.



ID	Full Name	Sess
391569012	Browning, Allan T.	FA96
345098754	Smith, Roxanne N.	FA96
591320941	Dobrowski, George S.	FA96
783490100	Jennings, Christina A.	SP97
840917892	Brown, Garrett L.	FA96
955712309	Cummings, Charles C.	SP97

What Is a Jenzabar CX Table?

CX makes name distinctions in the usage of database tables. A *table* in the CX contains information that remains static and is denoted with the *_table* extension. For example, the State table, named *st_table,* contains the list of states in the United States of America. On the CX menu, you can access most tables in *Table Maintenance* menus.

What Is a Jenzabar CX Record?

CX makes name distinctions in the usage of database tables. A *record* in the CX is a table that contains information that changes on a regular basis and is denoted with the *_rec* extension. For example, the Alternate Address record, named aa*_rec*, contains any other addresses at which students can be contacted, such as a summer address. You access records in CX program screens, scroll screens, and PERFORM screens.

Common Tables and Records

Several common tables are used in programs in the Purchasing and Accounts Payable product, as well as in other products throughout CX. This includes the following common tables:

- Alternate Address record
- Building table
- Country table
- Facility table
- ID record
- State table
- User ID table

General Ledger Tables and Records

Purchasing and Accounts Payable uses the following General Ledger tables and records:

- Document table (doc_table)
- General Ledger Account record (gla_rec)
- General Ledger Amount record (gl_amt_rec)
- General Ledger Entry record (gle_rec)
- General Ledger Permission table (glperm_table)
- General Ledger Substitution table (glsub_table)
- General Ledger Transaction record (gltr_rec)
- Journal (Voucher) record (vch_rec)
- Subsidiary Account record (suba_rec)
- Subsidiary Balance table (subb_table)
- Subsidiary Entry record (sube_rec)
- Subsidiary table (subs_table)
- Subsidiary Transaction record (subtr_rec)

Note: For more information about the General Ledger tables and records, see *General Ledger Technical Manual*.

Requisitioning Tables and Records

The following tables and records originate in the Requisitioning product and provide information for purchase orders, invoices or check requests in Purchasing and Accounts Payable:

- Account Authorization table (auth_acct_table)
- Approved Requisition Line Item record (appritem_rec)
- Commodity Code Authorization table (auth_comm_table)
- Denial Comment Entry record (denial_blob)
- ID Authorization table (auth_id_table)
- Pending Requisition record (pendreq_rec)
- Requisition Address Information record (reqaddr_rec)
- Requisition Header Comment record (reqhead_blob)
- Requisition Header Entries record (reqhead_rec)
- Requisition Line Item Account Entry record (reqacct_rec)
- Requisition Line Item Comment record (reqline_blob)
- Requisition Line Item record (reqline_rec)

Note: For more information about the Requisitioning tables and records, see *Requisitioning Tables and Records* in this manual.

Required Tables and Records

The following tables and records are required to run the features of the Purchasing and Accounts Payable product.

- Document table
- Fiscal Calendar record
- ID record
- User ID table
- Vendor record

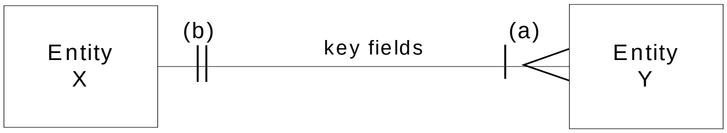
Table and Record Relationships

Key to Entity Relationship Diagrams

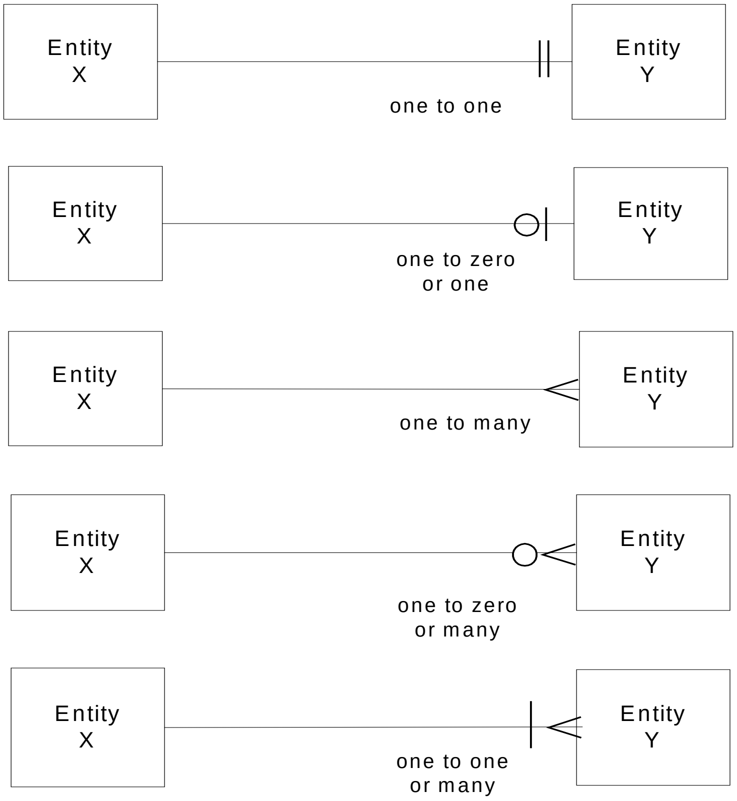
The diagrams on following pages show the relationship between the tables and records that the applications in the Purchasing and Accounts Payable product use. This diagram provides a key to interpreting the entity relationships in these diagrams.

Interpret the diagrams as follows:

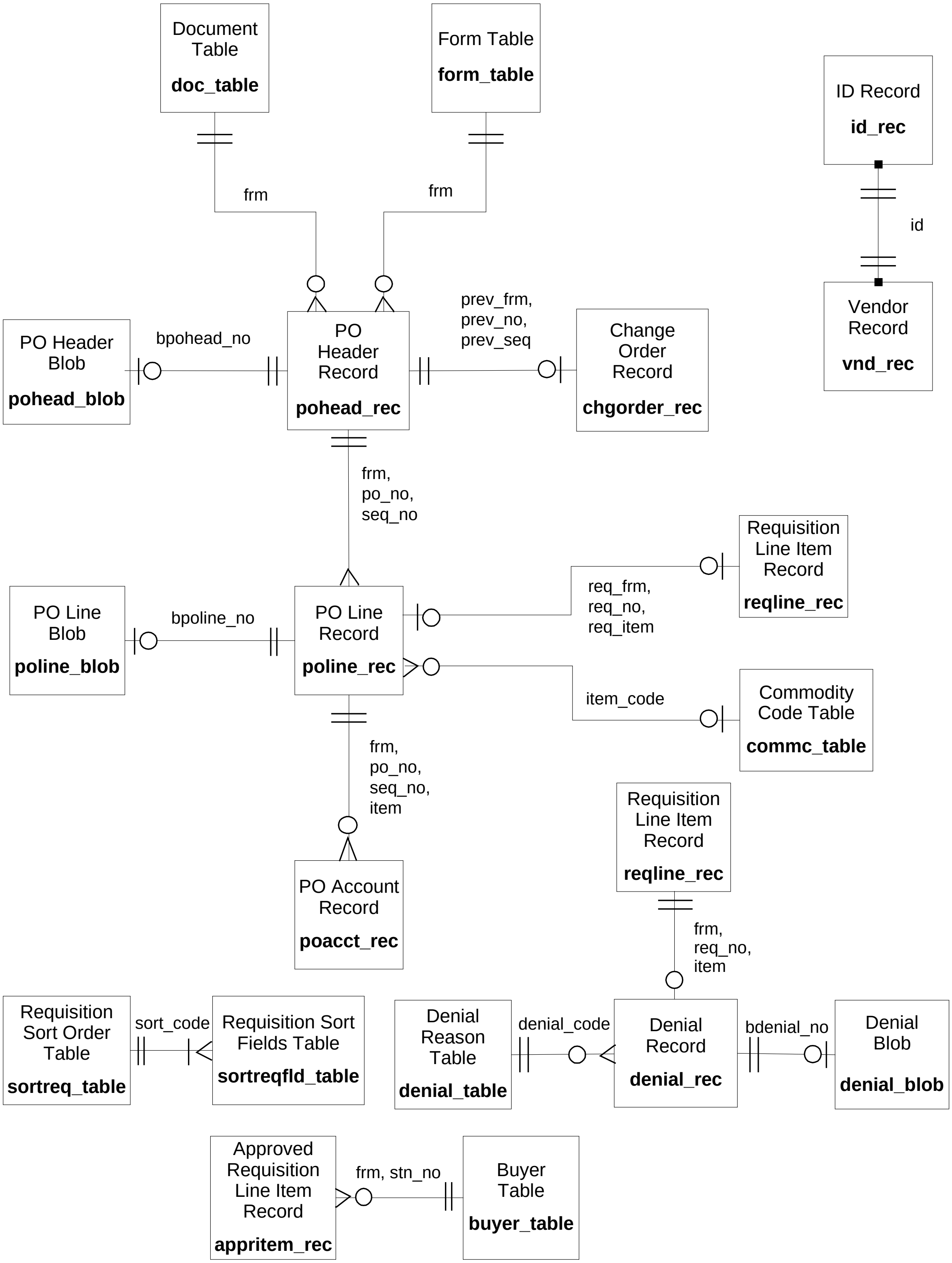
- Entity X has a one to (a) one or many relationship to Entity Y.
- Entity Y has a one to (b) one relationship to Entity X.



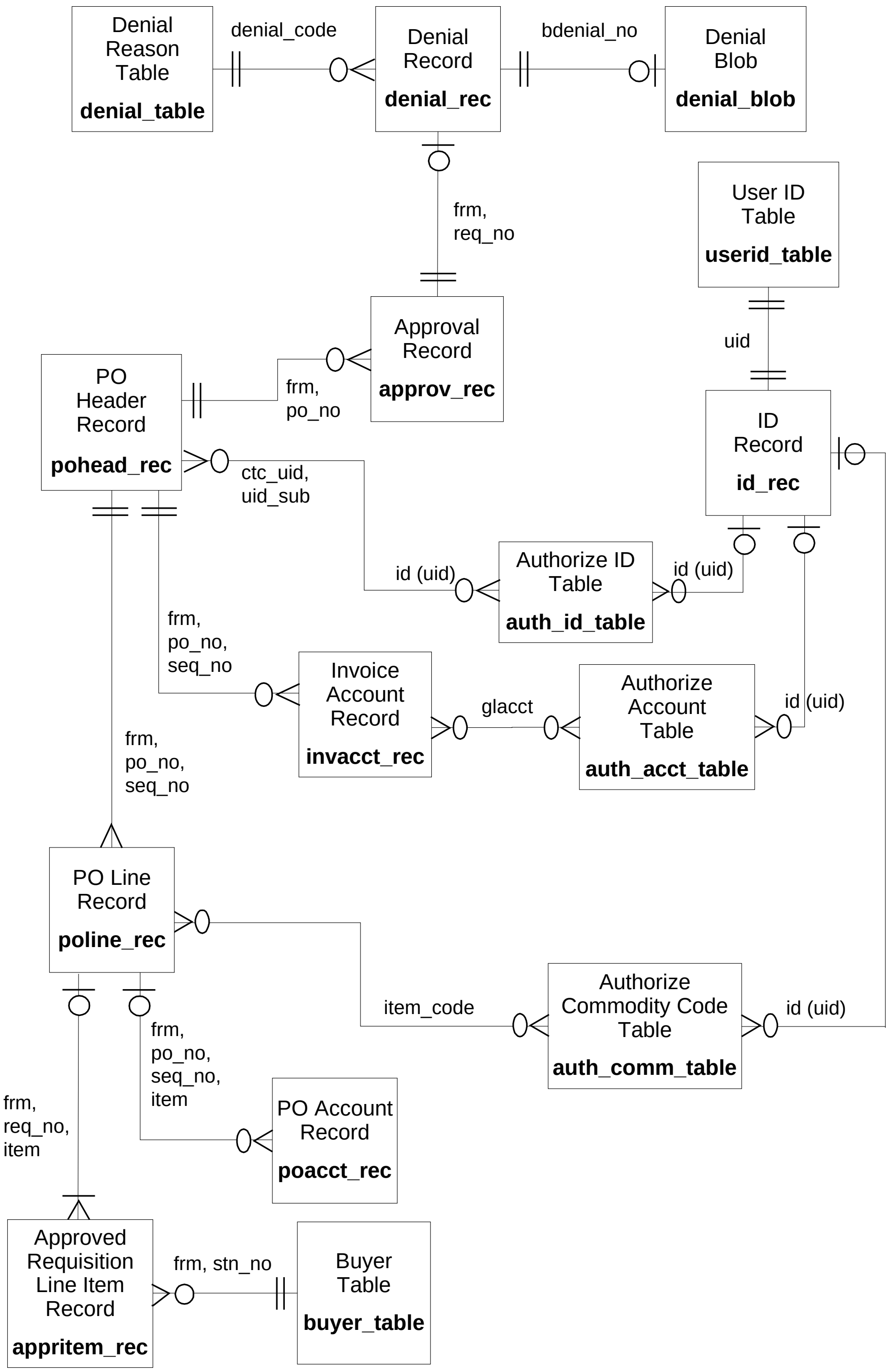
Relationship symbols and their meanings:



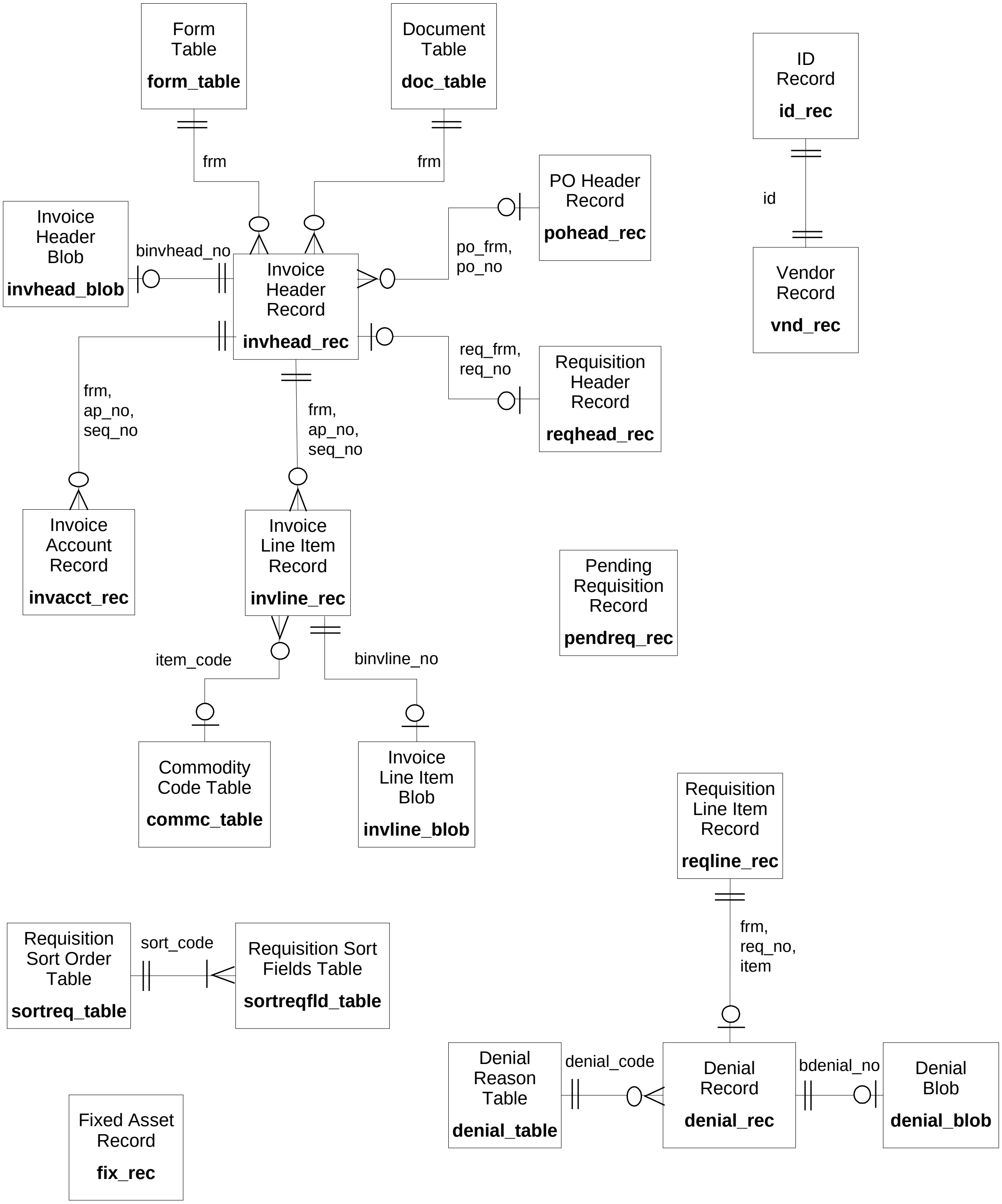
Purchase Application - Entity Relationship Diagram



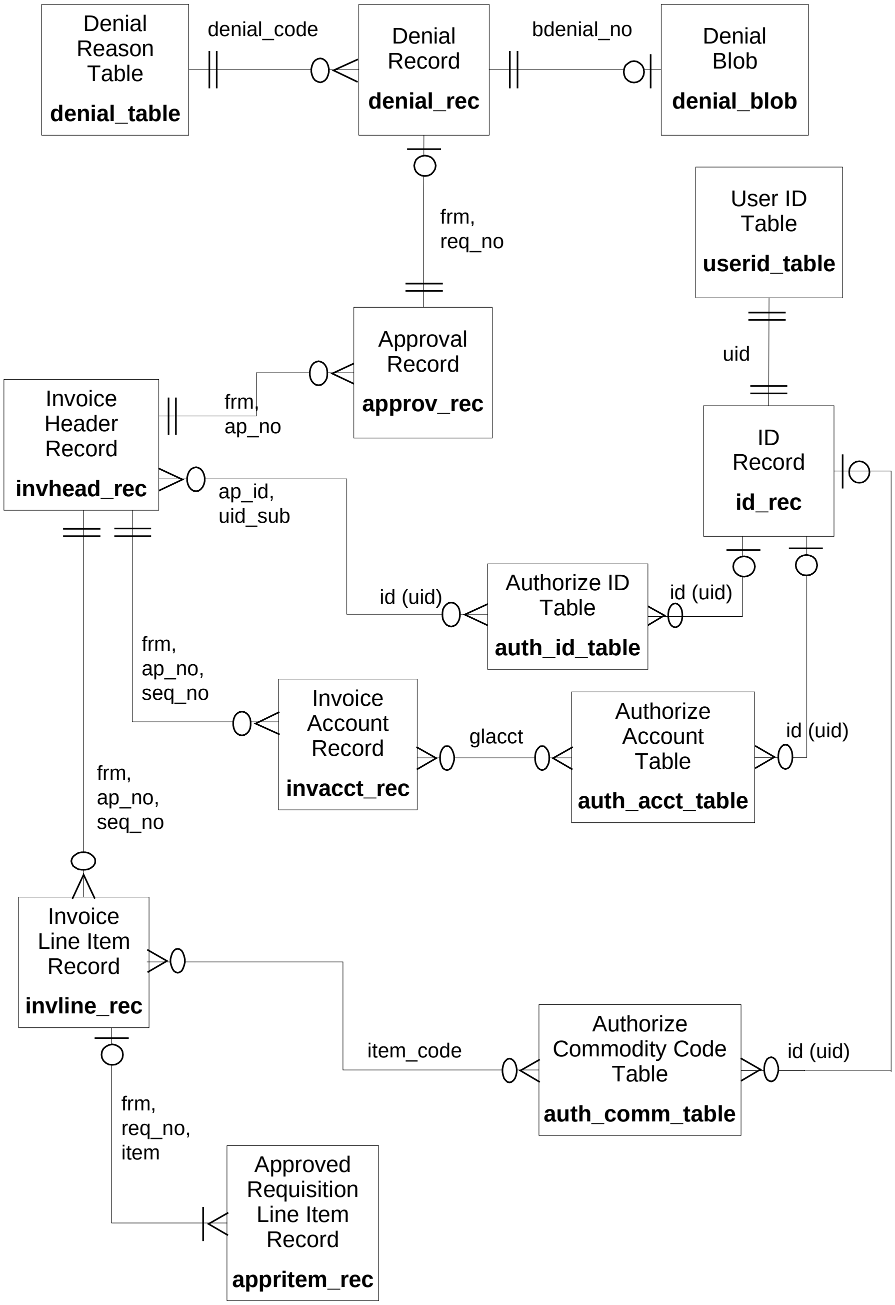
Purchase Order Approval - Entity Relationship Diagram



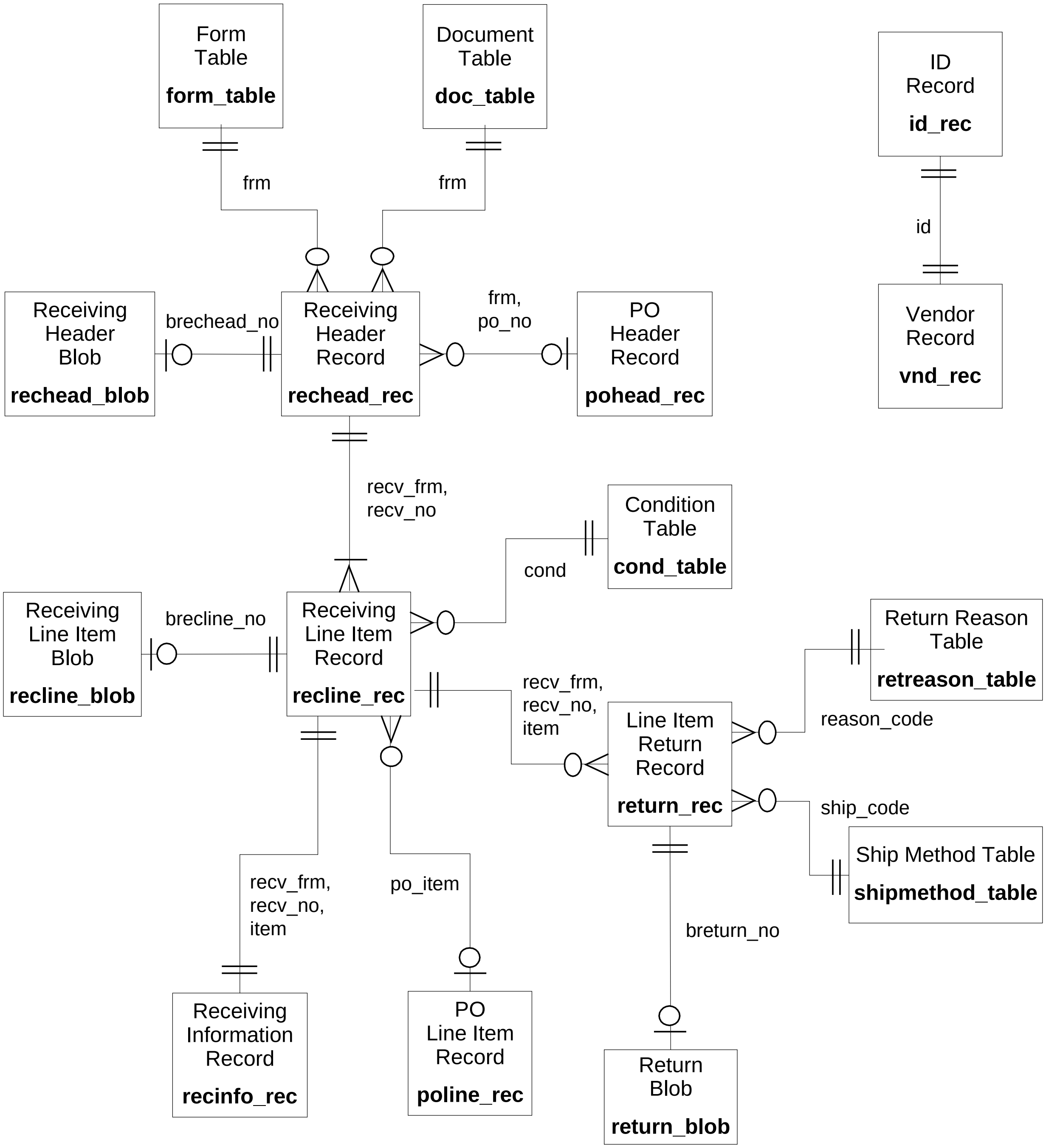
Accounts Payable Application - Entity Relationship Diagram



Accounts Payable Approval - Entity Relationship Diagram



Receiving Application - Entity Relationship Diagram



Purchasing and Accounts Payable Schemas

Introduction

Schema files define the structure of database files and associated fields in the CX data dictionary. You can access schema files associated with the Purchasing and Accounts Payable product in the following directory paths:

- $CARSPATH/schema/common
- $CARSPATH/schema/financial

Note: If you make changes to schemas for these tables and records, you will need to reinstall each associated product.

File Naming Conventions

CX makes name distinctions in the naming of schemas. For schema files containing definitions of CX tables, the UNIX file name begins with the letter *t* followed by characters describing the table's English name (e.g., tst for the State table). For schema files containing definitions of CX records, the UNIX file name describes the record's English name (e.g., as id for ID record).

The first line in a schema file, after revision information, specifies the INFORMIX database table that the schema defines. For example, *st_table* (State table) is specified in the *tst* schema file.

Field Descriptions

Schema files contain descriptions of each field defined in a table or record. You can view descriptions of fields in Purchasing and Accounts Payable tables and records by accessing the schema files

Schema File Reports

The standard CX includes three reports that provide information about the contents of the schema files. When table implementation begins, you can run the reports to provide the installation team with information about the tables and their fields:

Select the report options from the following menu path: System Management: Data Dictionary menu

The reports are as follows:

dbefield
Lists each column in the database by table, including its name, short and long descriptions, field type (e.g., char or date), and size

dbefile
Lists the tables that relate to each track area (e.g., ADM or COM), including the table name, description and purpose.

dbetrack
Combines the contents of dbefield and dbefile, displaying the tables for each track and the columns in each table.

Purchasing and Accounts Payable Tables and Records

Introduction

The following listing includes the primary tables and records that are used in the Purchasing and Accounts Payable product. This listing indicates each table/record's purpose and association with programs and other tables and records. The common tables and records are listed first, followed by the financial tables and records.

Note: In this listing, the listed programs are within the Purchasing and Accounts Payable product, and the listed products and applications are outside the Purchasing and Accounts Payable product.

Common Tables and Records

The following tables and records contain common, general information that several products or Jenzabar products use.

Alternate Address record

UNIX filename: aa

Informix filename: aa_rec

Schema location: $CARSPATH/schema/common

Purpose: Defines the alternate address information for an individual.

Program interrelationships: Accounts Payable, Mass Invoice, Purchasing, Receiving

Product interrelationships: Cashier, File Posting, Financial Aid, Lib, Personnel/ Payroll, Student Billing

Table/record interrelationships: id_rec

Building table

UNIX filename: tbldg

Informix filename: bldg_table

Schema location: $CARSPATH/schema/common

Purpose: Contains descriptions of known buildings and defines an identifying code for each.

Program interrelationships: Purchasing, Receiving

Product interrelationships: Events Management, Registration, Requisitioning

Table/record interrelationships: None

Country table

UNIX filename: tctry

Informix filename: ctry_table

Schema location: $CARSPATH/schema/common

Purpose: Contains code and country name for the countries known to the system.

Program interrelationships: Accounts Payable, Purchasing

Product interrelationships: Requisitioning

Table/record interrelationships: id_rec

Facility table

UNIX filename: tfacil

Informix filename: facil_table

Schema location: $CARSPATH/schema/common

Purpose: Describes the room facilities available and defines the campus and building where those rooms are located.

Program interrelationships: Receive

Product interrelationships: Purchasing, Registration, Requisitioning, Student Affairs and Housing

Table/record interrelationships: None

Identification record

UNIX filename: id

Informix filename: id_rec

Schema location: $CARSPATH/schema/common

Purpose: Contains the names and general information for all individuals and entities.

Program interrelationships: Used by all programs

Product interrelationships: Used by all products

Table/record interrelationships: aa_rec, profile_rec, vnd_rec

State table

UNIX filename: tst

Informix filename: st_table

Schema location: $CARSPATH/schema/common

Purpose: Defines the state codes and their corresponding names.

Program interrelationships: Accounts Payable, Mass Invoice, Purchasing

Product interrelationships: Used by all products

Table/record interrelationships: None

User Identification table

UNIX filename: tuserid

Informix filename: userid_table

Schema location: $CARSPATH/schema/common

Purpose: Provides a link between the system (login) user ID number and the CX database ID number.

Program interrelationships: Accounts Payable, Assign Buyer, Purchasing, Receiving

Product interrelationships: Used by all products

Table/record interrelationships: id_rec

Financial Tables and Records

The following tables and records contain information specifically used in accounting or financial applications in CX.

Buyer table

UNIX filename: tbuyer

Informix filename: buyer_table

Schema location: $CARSPATH/schema/financial

Purpose: Assigns a buyer to a station number for assigning requisitions.

Program interrelationships: Assign Buyer

Product interrelationships: None

Table/record interrelationships: doc_table

Commodity Code Entries table

UNIX filename: tcommc

Informix filename: commc_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the Commodity Code record associated with the Item Code and contains a Central Store Availability flag.

Program interrelationships: Accounts Payable, Purchasing, Requisition

Product interrelationships: Requisitioning

Table/record interrelationships: invline_rec, poline_rec, reqline_rec

Commodity table

UNIX filename: tcommod

Informix filename: commod_table

Schema location: $CARSPATH/schema/financial

Purpose: Describes the types of commodities charged in an invoice.

Program interrelationships: Mass Invoice

Product interrelationships: None

Table/record interrelationships: None

Denial Code table

UNIX filename: tdenial

Informix filename: denial_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the Denial Codes and their corresponding reasons for denying a requisitioned line item in the Approval Process.

Program interrelationships: Accounts Payable, Purchasing

Product interrelationships: Requisitioning

Table/record interrelationships: linedenial_rec

Fiscal Calendar record

UNIX filename: fsclcal

Informix filename: fscl_cal_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains valid fiscal posting periods and their valid beginning/ending and opening/closing dates.

Program interrelationships: Mass Invoice, Purchasing

Product interrelationships: All Financial products and applications

Table/record interrelationships: vch_rec

Fixed Asset record

UNIX filename: fix

Informix filename: fix_record

Schema location: $CARSPATH/schema/financial

Purpose: Contains information about fixed assets, including location, depreciation details, and authorization.

Program interrelationships: Accounts Payable, Purchasing

Product interrelationships: Fixed Assets, General Ledger

Table/record interrelationships: fix_table

Form Type table

UNIX filename: tform

Informix filename: form_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the various form types used by the Requisitioning, Purchasing and Accounts Payable products.

Program interrelationships: Accounts Payable, Purchasing, Receiving, Requisition

Product interrelationships: Requisitioning

Table/record interrelationships: invhead_rec, pohead_rec, rechead_rec, reqhead_rec

General Ledger Substitution table

UNIX filename: tglsub

Informix filename: glsub_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the segments of the General Ledger Account number that trigger the Account Auto-Fill feature.

Program interrelationships: Accounts Payable, Approval, Mass Invoice, Purchasing, Recurrent, Requisition

Product interrelationships: General Ledger, Requisitioning

Units table

UNIX filename: tunits

Informix filename: units_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the codes and corresponding descriptions used to identify the type of packaging (e.g., boxes, cases, each) specified for purchase order, requisitioning and receiving line items.

Program interrelationships: Accounts Payable, Purchasing, Receiving, Requisition

Product interrelationships: Requisitioning

Table/record interrelationships: None

Purchasing and Accounts Payable Tables

The following tables and records originate or have primary use in Purchasing and Accounts Payable:

Invoice Account Entries record

UNIX filename: invacct

Informix filename: invacct_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains account information associated with specified invoices' line items

Program interrelationships: Accounts Payable

Product interrelationships: None

Table/record interrelationships: invline_rec

Invoice Category table

UNIX filename: tinvctgry

Informix filename: invctgry_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the categories of invoices and their descriptions.

Program interrelationships: Mass Invoice

Product interrelationships: None

Table/record interrelationships: invoice_rec

Invoice Charge record

UNIX filename: invchg

Informix filename: invchg_rec

Schema location: $CARSPATH/schema/financial

Purpose: Maintains records of accounts that have been charged for invoices.

Program interrelationships: Mass Invoice

Product interrelationships: None

Table/record interrelationships: invoice_rec

Invoice Header Comment record

UNIX filename: binvhead

Informix filename: invhead_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the specified invoices.

Program interrelationships: Accounts Payable

Product interrelationships: None

Table/record interrelationships: invhead_rec

Invoice Header Entries record

UNIX filename: invhead

Informix filename: invhead_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains the header information for specified invoices, including the form type and number, the document number, the status, terms, and due date of the invoice, as well as payee and amount information.

Program interrelationships: Accounts Payable

Product interrelationships: None

Table/record interrelationships: doc_table, invhead_blob, invline_rec

Invoice Line Item Comment record

UNIX filename: binvline

Informix filename: invline_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the invoice line item.

Program interrelationships: Accounts Payable

Product interrelationships: None

Table/record interrelationships: invline_rec

Invoice Line Item Entries record

UNIX filename: invline

Informix filename: invline_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains the detail information for line items on an invoice (e.g., quantity, price, and description).

Program interrelationships: Accounts Payable

Product interrelationships: None

Table/record interrelationships: commc_table, invacct_rec, invhead_rec, invline_blob

Invoice record

UNIX filename: invoice

Informix filename: invoice_rec

Schema location: $CARSPATH/schema/financial

Purpose: For each invoice, indicates the amount, discount and mailing address.

Program interrelationships: Mass Invoice

Product interrelationships: None

Table/record interrelationships: invchg_rec

Invoice Subsidiary record

UNIX filename: invsubs

Informix filename: invsubs_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains subsidiary information for line item charges in an invoice.

Program interrelationships: Mass Invoice

Product interrelationships: None

Table/record interrelationships: invoice_rec

Line Item Return Entries record

UNIX filename: return

Informix filename: return_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines the receiving line item return information.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: recinfo_rec, retreason_table

Method of Shipping Returned Goods table

UNIX filename: tshpmeth

Informix filename: shipmethod_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the types of shipping methods used when returning goods in the Receiving application.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: return_rec

Payment Terms table

UNIX filename: tpayterm

Informix filename: pay_term_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines valid payment terms and gives valid discounts for each term.

Program interrelationships: Accounts Payable, Mass Invoice

Product interrelationships: General Ledger

Table/record interrelationships: suba_rec, subs_table

Purchase Order Header Comment record

UNIX filename: bpohead

Informix filename: pohead_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the Purchase Order Header Entry record.

Program interrelationships: Purchasing

Product interrelationships: None

Table/record interrelationships: pohead_rec

Purchase Order Header Entry record

UNIX filename: pohead

Informix filename: pohead_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains the general information about a purchase order (e.g., dates, vendor numbers, and amounts).

Program interrelationships: Accounts Payable, Purchasing, Receiving

Product interrelationships: None

Table/record interrelationships: doc_table, form_table, pohead_blob, poline_rec

Purchase Order Line Item Account Entries record

UNIX filename: poacct

Informix filename: poacct_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines purchase order line item account data (e.g., the account, amount, and subsidiary information).

Program interrelationships: Accounts Payable, Purchasing

Product interrelationships: None

Table/record interrelationships: poline_rec

Purchase Order Line Item Comment record

UNIX filename: bpoline

Informix filename: poline_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the Purchase Order Line Item Entries record.

Program interrelationships: Purchasing

Product interrelationships: None

Table/record interrelationships: poline_rec

Purchase Order Line Item Entries record

UNIX filename: poline

Informix filename: poline_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains detail information about the item on the purchase order (e.g., quantity, unit cost, discount, description).

Program interrelationships: Accounts Payable, Purchasing, Receiving

Product interrelationships: None

Table/record interrelationships: poacct_rec, pohead_rec

Purchase Order record

UNIX filename: po

Informix filename: po_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains information relating to purchase orders, including amounts, dates and vendors.

Program interrelationships: Mass Invoice

Product interrelationships: None

Table/record interrelationships: None

Reason for Returning Goods table

UNIX filename: tretreas

Informix filename: retreason_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the codes and corresponding descriptions used to identify the reason line items were returned.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: return_rec

Receiving Header Comment record

UNIX filename: brechead

Informix filename: rechead_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the Receiving Header Entries record.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: rechead_rec

Receiving Header Entries record

UNIX filename: rechead

Informix filename: rechead_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines the receiving header information specifying a packing slip number and the associated description.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: recinfo_rec, recline_rec

Receiving Line Item Comment record

UNIX filename: brecline

Informix filename: recline_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for the Receiving Line Item Entries record.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: recinfo_rec, recline_rec

Receiving Line Item Condition table

UNIX filename: tcond

Informix filename: cond_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the codes that identify the condition of line items received.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: recline_rec

Receiving Line Item Entries record

UNIX filename: recline

Informix filename: recline_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines the Receiving line item information, defining the condition of the received line item and the quantity received.

Product interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: cond_table, rechead_rec, retreason_rec, return_rec, shipmethod_table

Receiving Line Item Info record

UNIX filename: recinfo

Informix filename: recinfo_rec

Schema location: $CARSPATH/schema/financial

Purpose: Defines the Receiving line item data for non-purchase order line items.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: cond_table, rechead_rec, retreason_rec, return_rec, shipmethod_rec

Receiving Return Comment record

UNIX filename: breturn

Informix filename: return_blob

Schema location: $CARSPATH/schema/financial

Purpose: Provides the free form text comments for merchandise that the institution returns to the vendor.

Program interrelationships: Receiving

Product interrelationships: None

Table/record interrelationships: return_rec

Requisition Line Item Denial Entries record

UNIX filename: ldenial

Informix filename: linedenial_rec

Schema location: $CARSPATH/schema/financial

Purpose: Tracks the requisition line items that buyers at the institution have denied.

Program interrelationships: Approval, Purchasing

Product interrelationships: Requisitioning

Table/record interrelationships: denial_blob, denial_rec, recline_rec

Requisition Sort Fields table

UNIX filename: tsrtrfld

Informix filename: sortreqfld_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the fields to be used and their sort order for sorting requisitions and purchase orders.

Program interrelationships: Accounts Payable, Assign, Purchasing

Product interrelationships: Requisitioning

Table/record interrelationships: appritem_table, sortreq_table

Requisition Sort Order table

UNIX filename: tsrtr

Informix filename: sortreq_table

Schema location: $CARSPATH/schema/financial

Purpose: Defines the codes and corresponding descriptions used to identify the order for sorting requisitions and purchase orders.

Program interrelationships: Accounts Payable, Assign, Purchasing

Product interrelationships: Requisitioning

Table/record interrelationships: appritem_rec, sortreqfld_table

Vendor record

UNIX filename: vnd

Informix filename: vnd_rec

Schema location: $CARSPATH/schema/financial

Purpose: Contains information about vendors (e.g., name, contact person, delivery terms)

Program interrelationships: Accounts Payable, Purchasing

Product interrelationships: Requisitioning

Table/record interrelationships: id_rec, aa_rec

Vendor Quality table

UNIX filename: tvenqual

Informix filename: venqual_table

Schema location: $CARSPATH/schema/financial

Purpose: Contains information about special characteristics of vendors (e.g., minority ownership).

Program interrelationships: Vendor Entry

Product interrelationships: None

Table/record interrelationships: id_rec, vnd_rec

Vendor Type table

UNIX filename: tventype

Informix filename: ventype_table

Schema location: $CARSPATH/schema/financial

Purpose: Contains information about the nature of vendors (e.g., sole proprietorship, distributor of maintenance equipment, or other user-defined information about vendors).

Program interrelationships: Vendor Entry

Product interrelationships: none

Table/record interrelationships: id_rec, vnd_rec

SECTION 6 - REQUISITIONING MACROS AND INCLUDES

Overview

Introduction

This section of Jenzabar technical manuals provides you with reference information about the macros and includes used to set up a product.

The Relationship Among Macros, Includes, and C Programs

For all elements of the product other than C program, m4 macros contain the definitions of features that have been designed to be modified for each institution. These macros, located under $CARSPATH/macros, are passed throught the m4 processor.

CX contains an alternative method for the setting of features in C programs. Macros in the source code of C programs are not passed through the m4 processor because the C compiler has its own preprocessor, the cc.

To provide the same apparent functionality to C programs, a section in the include source directory has been allocated for a type of include file that acts as an intermediary between the m4 processor and the cc preprocessor. In operation, m4 macros are defined whose output is a valid cc macro. These m4 macros are placed in the include files. Then the include files are translated and the appropriate cc macro is placed in the include file. The installed include file is included by the C compiler at compile time so that the result of the compilation is influenced by the m4 macro.

General Installation Procedures

See *CX Technical Manual* for general procedures on setting and installing changes to macros and includes.

Configuration Table

The Configuration table (config_table) will eventually replace macros for every CX product. Within Requisitioning, the following Configuration table entries currently exist:

ENABLE_APPROVAL_BLOB_ACCESS
Enables requisition approvers to view the entire contents of a Binary Large Object (BLOB) field associated with a requisition. If enabled with a Y in the Value field of the Configuration table, the approver will be able to display the entire BLOB by clicking on it. If disabled with an N, the approver will be required to print the requisition to read the complete BLOB field.

ENABLE_DEPT_BGT_CHECK
Allows each department to do budget checking at their level of the account structure as well as at the normal budgeted level. If enabled with a Y in the Value field of the Configuration table, budget checking at the department level will be enabled.

ENABLE_BUDGET_CHECK
Used by the *purchase*, *acctspay*, *massinv*, and *requisition* programs, enables an institution to eliminate budget checking. If enabled with a Y in the Value field of the Configuration table, budget checking will occur in these four programs. If set to N, the program will perform no budget checks. The default is Y.

ENABLE_PEND_REQ

Controls the encumbering of pending requisitions. If enabled with a Y in the Value field of the Configuration table, the screens and reports that display encumbered amounts will also include pending requisition amounts. By encumbering pending amounts, end users can view the financial status of an account and determine if sufficient funds will exist for a purchase *even if* all existing requisitioned amounts are approved and purchased. As delivered in standard CX, this feature is enabled.

ENABLE_REQ_ACCOUNT_CHECK

Controls the ability for users to associate unused but valid accounts with requisitions (i.e., accounts that are valid according to the Defined Accounts record [gld_rec] but for which no General Ledger Account record [gla_rec] exists). If set to N, the *requisition* program will not check for a gla_rec and will permit the user to enter any valid account with a requisition. If set to Y, *requisition* will require the user to enter a previously used account with a requisition.

Enable Macros for Requisitioning

The m4_define macro ENABLE_FEAT_REQUISITION enables users to use Requisitioning and to view Requisition and Approval features on menus. The default value for the macro is Y. To disable the Requisitioning product and menu options, set the value to N.

Other Macros and Includes

The other macros and includes that impact Requisitioning also affect the Purchasing and Accounts Payable product. For more information about these macros and includes, see *Purchasing and Accounts Payable Macros and Includes* in this manual*.*

SECTION 7 - PURCHASING AND ACCOUNTS PAYABLE MACROS AND INCLUDES

Overview

Introduction

This section provides reference information about the macros and includes used to set up the Purchasing and Accounts Payable product.

The Relationship Among Macros, Includes, and C Programs

For all elements of the product other than C program, m4 macros contain the definitions of features that have been designed to be modified for each institution. These macros, located under $CARSPATH/macros, are passed throught the m4 processor.

CX contains an alternative method for the setting of features in C programs. Macros in the source code of C programs are not passed through the m4 processor because the C compiler has its own preprocessor, the cc.

To provide the same apparent functionality to C programs, a section in the include source directory has been allocated for a type of include file that acts as an intermediary between the m4 processor and the cc preprocessor. In operation, m4 macros are defined whose output is a valid cc macro. These m4 macros are placed in the include files. Then the include files are translated and the appropriate cc macro is placed in the include file. The installed include file is included by the C compiler at compile time so that the result of the compilation is influenced by the m4 macro.

General Installation Procedures

See the *CX Implementation and Maintenance Technical Manual* for general procedures on setting up and installing changes to macros and includes.

Configuration Table

The purpose of the Configuration table (config_table) is to define default values and to implement features in CX. This table replaces macros for some CX programs. The following Configuration table entries relate to Purchasing and Accounts Payable:

ENABLE_DEPT_BGT_CHECK
Allows each department to do budget checking at their level of the account structure as well as at the normal budgeted level. If enabled with a Y in the Value field of the Configuration table, budget checking at the department level will be enabled.

ENABLE_PEND_REQ
Controls the encumbering of pending requisitions. If enabled with a Y in the Value field of the Configuration table, the screens and reports that display encumbered amounts will also include pending requisition amounts. By encumbering pending amounts, end users can view the financial status of an account and determine if sufficient funds will exist for a purchase *even if* all existing requisitioned amounts are approved and purchased. As delivered in standard CX, this feature is enabled.

ENABLE_RPA_ACCT_CONTROL
Used in the *acctspay* program to validate that the ACT amount type General Ledger accounts entered on an invoice are within the Major range of the ENC amount type account. This macro satisfies a budget control requirement for the state of Illinois and may be applicable to other states or institutions.

ENABLE_RPA_CLASS_CONTROL

Used in the *acctspay* program account validation process. This table entry verifies that, if the General Ledger account used to encumber a PO contains a classification (e.g., I or E), then the ACT account used to post the invoice contains the same classification code.
Note: For this enable table entry to operate correctly, the ENABLE_RPA_ACCT_CONTROL macro must also be defined with Y. In addition, the General Ledger account field used to contain the classification code field must be defined for *acctspay*. For more information, see *Customizing the Purchasing and Accounts Payable Processes* in this manual.

RPA_SUBS_AP_VALID

Used by the *poinvaudit* program to identify valid subsidiaries that can be used as control accounts in purchasing and accounts payable. Values must match the values defined in $CARSPATH/nmacros/custom/financial. This macro is necessary for *poinvaudit* to run successfully.
Example: A/P + PIP + PIPC
Note: You must follow the format as depicted. For the parsing purposes of this statement, data *must* be entered in the same format as the example.

Macros

Introduction

CX contains macros that define specific values and cause some special features to become available on menus. The macros enable you to change the available values, options and functionality of the system without having to modify C code. By modifying macros, you can customize your implementation of the CX, and make it easier to maintain.

Definition and Function

A macro is an instruction that causes the execution of a pre-defined sequence of instructions in the same source language. A macro consists of uppercase letters and underscores, and is used in place of a text string within source files. CX expands the macro to the longer text during the installation process. CX uses the following kinds of macros:

- Enable - allows you to enable a CX feature
- DBS_COMMON - allows you to define database values in screens
- Periodic - allows you to make changes on a periodic basis
- Static - allows permanent definition of a widely used value or comment

Macros can perform one of the following functions:

- Define commonly used phrases
- Define defaults on a screen (_DEF)
- Define valid values in a field (_VALID or _INCL)
- Enable system products (ENABLE_MOD)
- Enable system features (ENABLE_FEAT)
- Establish a valid value for an include

In many cases, the Purchasing and Accounts Payable product references the macros defined during the implementation of the General Ledger product. For more information about the macros, see *General Ledger Technical Manual*.

How to Locate Macros

The macros that affect Purchasing and Accounts Payable appear in the following macro file:

$CARSPATH/macros/custom/financial

Purchasing and Accounts Payable Enable Macros

The following macros enable features within Purchasing and Accounts Payable:

m4_define(‘ENABLE_FEAT_ASSIGN’,’Y’)
When set to the default value “Y”, enables the Assign Buyer (*assign*) application. Within the Purchasing and Accounts Payable product, enables a buyer to see only the approved requisitions assigned to him/her.

m4_define(‘ENABLE_FEAT_FAST_INVOICE,’Y’)
When set to the default value “Y”, enables users to select the Fast Invoice command within Invoice Entry (acctspay), providing access to Simplified Invoice processing.

m4_define(‘ENABLE_FEAT_PURCHASE’,’Y’)
When set to the default value “Y”, enables the Purchasing (*purchase*) application, permitting users to view and use the Purchasing features from their menus.

m4_define(‘ENABLE_FEAT_VENDOR_CHECK’,’Y’)

When set to the default value “Y”, requires that only existing vendors can be specified on purchase orders. This feature protects the institution from the use of unauthorized vendors.

Requisition **Define Macros**

The following macros define valid values used within the *requisition* program:

m4_define (‘REQ_CHK_DOC_REF’,’RK’)
m4_define (‘REQ_PO_DOC_REF’,’RP’)
m4_define (‘REQ_INV_DOC_REF’,’RI’)

Define the valid requisition form types.

Note: The following setup considerations exist for these macros:

- The requisition form types must be valid in the Document table (doc_table) for station 0 and for any other desired stations.
- They must also be valid in the Form table (form_table), with an Application Code (app_code) of RQ.
- If the institution is using the Assign Buyer feature, then the forms for REQ_PO_DOC_REF (i.e., RP) and REQ_INV_DOC_REF (i.e., RI) must be valid in the Buyer table.

m4_define (‘RPA_QTY_AS_FLOAT’,’N’)

Causes quantity fields on requisition, purchase order, and invoice screens and reports to display as floating decimals. The standard definition for the macro is N, causing the quantity fields to accept and display whole numbers only.

Purchase **and** ***Acctspay*** **Define Macros**

The following macros define valid values used within the *purchase* program and/or the *acctspay* program:

m4_define (‘TM_DOC_REF’,’TM’)

Defines the temporary purchase order and temporary invoice form type that the system uses to create either purchase orders or invoices from requisitions. The TM form type must be valid in the Document table (doc_table) for station 1.

m4_define (‘RPA_CLOSE_PO_DEFAULT’,’N’)

Defines the default value that displays in the Close PO field on the Invoice Header screen. If set to Y, *acctspay* will close a purchase order and relieve all associated encumbrances when the invoice is submitted.

m4_define(`RPA_SHIPTO_DEFAULT',`0')

If you use centralized receiving, defines the ID number of the central receiving location at your institution. This macro causes the defaulting of the central receiving address in Purchase Order Entry. If set to 0, no default address will display in Purchase Order Entry.

Massinv **Define Macro**

The following macro defines a valid value used within the *massinv* program:

m4_define (‘DEFAULT_FUND’,’10’)

Defines the default fund code for the Mass Invoice application.

Acctspay **Define Macros**

The following macros define valid values used within the *acctspay* program.

m4_define (‘INV_DOC_REF’,’IV’)

m4_define (‘CM_DOC_REF’,’CM’)

Defines the invoice form type and credit memo form types.

Note: The following setup considerations exist for these macros:

- The form types must be valid in the Document table (doc_table) for station 1, and other stations if desired.
- The form types must also be valid in the Form table (form_table) with the Application Code (app_code) of AP.
- The requisition form REQ_CHK_DOC_REF can be used with INV_DOC_REF.

Receive Define Macro

The following macro defines a valid value used within the *receive* program:

m4_define (‘REC_DOC_REF’,’RV’)

Defines the default Receiving form types.

Note: The following setup considerations exist for this macro:

- The form type must be valid in the Document table (doc_table) for station 1, and other stations if desired.
- The form type must also be valid in the Form table (form_table) with the Application Code (app_code) of RC.

Purchase Define Macro

The following macro defines a valid value used within the Purchasing program:

m4_define (‘PO_DOC_REF’,’PO’)
m4_define (‘PP_DOC_REF’,’PP’)
m4_define (‘PI_DOC_REF’,’PI’)
m4_define (‘PB_DOC_REF’,’PB’)

Define purchase order default form types.

Note: The following setup considerations exist for these macros:

- The purchase order form types must be valid in the Document table (doc_table) for station 0 and for any other desired stations.
- They must also be valid in the Form table (form_table), with an Application Code (app_code) of PU.
- All purchase order form types must have an associated change order form type in the Document table (doc_table). Purchase order form types and associated change order form types use the same second letter in their names. For example, the purchase order form type of PP has an associated change order form type of CP.

Relationships Between Form Types

The applications within Requisitioning, Purchasing and Accounts Payable use form types to control the types of processing performed on certain types of transactions. For a complete description of the relationships among programs, forms and processes, see *Purchasing and Accounts Payable Processes* in this manual.

The programs assume specific relationships between the form types, as shown on the following list:

- Between requisition and purchase order form types:
  - REQ_PO_DOC_REF (default value RP) can be used with PO_DOC_REF (open PO)
  - REQ_PO_DOC_REF (default value RP) can be used with PP_DOC_REF (PO)
  - REQ_PO_DOC_REF (default value RP) can be used with PB_DOC_REF (blanket PO)
  - REQ_INV_DOC_REF (default value RI) can be used with PI_DOC_REF (PO for invoice)
- Between requisition for check and invoice form type:
  - REQ_CHK_DOC_REF (default value RK) can be used with INV_DOC_REF (invoice)

Note: If the institution is using the Assign Buyer feature, you must add entries to the Buyer table to associate station/form type combinations with the desired requisition type.

Includes

Introduction

CX contains includes that determine the features that are enabled in the products. An include can either be a compile option that enables or disables a feature, or a default value include that defines a default value for a feature.

To enable a feature in the Purchasing and Accounts Payable product, you must define an include in *$CARSPATH/include/custom*. To disable an include, comment out the include in the same file. See *CX Technical Manual* for more information on enabling and disabling includes. By modifying includes, you can customize your implementation of the Purchasing and Accounts Payable product, and make the product easier to maintain.

Purpose

An include allows you to activate or deactivate features in C programs without changing the C code.

Macro Dependency

Includes have a dependency on macros. Normally, you do not directly modify includes for the product. You must modify corresponding macro values and then reinstall the include.

How to Locate Includes

Includes that affect the Purchasing and Accounts Payable product appear in the following file:
$CARSPATH/include/custom/rpa

Purchasing and Accounts Payable Includes

The following includes relate to Purchasing and Accounts Payable. When you implement the product, you do not need to modify any of these includes; they are listed here for reference only.

```
/*-----
   If ENABLE_FEAT_ASSIGN is set to "Y", the purchase application will limit
        the view of approved requisitions to only those assigned to a buyer
        via station number.
-----*/
m4_keepif(ENABLE_FEAT_ASSIGN, ~`Y~')
#define RPA_ASSIGN
m4_keepend

/* -----
       If ENABLE_FEAT_VENDOR_CHECK is set to "Y", the Vendor will be force
       checked in the purchase and acctspay applications.
----- */
m4_keepif(ENABLE_FEAT_VENDOR_CHECK, ~`Y~')
#define VENDOR_CHECK
m4_keepend

/* -----
       Types of bgvoucher transactions.
----- */
#define ENC_ADD_CHG_TYPE         "EACG"
#define ENC_ADD_INV_TYPE         "EAIV"
#define ENC_UPD_CHG_TYPE         "EUCG"
#define ENC_UPD_INV_TYPE         "EUIV"
#define MANUAL_INV_TYPE          "INV "
#define ACT_ADD_INV_TYPE         "AINV"
#define ACT_UPD_INV_TYPE         "UINV"
#define ACT_CR_MEMO_TYPE         "CMEM"

/* -----
       Define the form types based on the values set up in the
       $CARSPATH/macros/custom/financial file.
----- */
#define REQ_PO_DOC_REFER         "REQ_PO_DOC_REF"
#define REQ_CHK_DOC_REFER        "REQ_CHK_DOC_REF"
#define REQ_INV_DOC_REFER        "REQ_INV_DOC_REF"
#define PO_DOC_REFER             "PO_DOC_REF"
#define PP_DOC_REFER             "PP_DOC_REF"
#define PI_DOC_REFER             "PI_DOC_REF"
#define PB_DOC_REFER             "PB_DOC_REF"
#define TM_DOC_REFER             "TM_DOC_REF"
#define INV_DOC_REFER            "INV_DOC_REF"
#define CM_DOC_REFER             "CM_DOC_REF"
#define REC_DOC_REFER            "REC_DOC_REF"

/* -----
       Default fund to appear when entering in accounts in
       the invoice application, aka Batch Invoice Entry.
----- */
#define DEFAULT_FUND_NAME        "DEFAULT_FUND"
```

SECTION 8 - JENZABAR CX SYSTEM PROGRAM FILES

Overview

Introduction

This section provides you with reference information about the files that relate to most CX programs. By understanding the file structure and the contents of the files, you can locate most of the information you need about any program.

Program Files Detailed

This section contains details about the following files:

def.c

The def.c file contains the declaration of external variables (including structures) that must be available to all source files in the program. You can also initialize these variables in this file. As with other C source files, the files also contain comments. The makedec command uses the def.c file to create the dec.h file.

mac.h

The mac.h file contains preprocessor include and define statements, typedef statements, and structure template definition statements. The file also contains macro substitution defines and declarations of structures. This file is included in all source files during compilation through use of the dec.h file.

Note: All other files for each CX program are standard C programming files with standard components and structure.

File Locations

All program files, including the def.c file and the mac.h file, reside in a source (src) directory in the following directory path: $CARSPATH/src/<product name>/<program name>. For example, the program files for the Accounts Payable program appear in $CARSPATH/src/purchase/acctspay.

Definition File

Every program uses a definition (def.c) file. The def.c file for a screen-oriented program can contain the following information:

- Includes for a mac.h file
- Declaration of global variables and structures used throughout the application
- Structure and non-structure screen binds (i.e., program buffer to screen buffer binds)
- Ring menu definitions
- Prompt line information
- Program parameters
- Declarations of dynamic memory (dmms, dmls, and dmlts) in relation to functionality within libdmm (the dynamic memory management package)
- Screen pointers

The def.c file for a non-screen-oriented program can contain the following information:

- Includes for a mac.h file
- Global program variables
- Includes for schema files' def.c files
- Form pointers that provide the location for forms
- Sqlda pointers that bind the file structure to the form
- dmm, dml and dmlt definitions
- Program parameters

- Declarations of functions so the compiler can handle a call of that function

Example of a def.c File

The following is an edited excerpt from the def.c file for the Purchasing and Accounts Payable program *purchase*. It appears here to illustrate the common components of a standard CX def.c file.

Note: The legend for the numbers at the beginning of each file component appears at the end of the excerpt.

```
1

#include "mac.h"

2

#include <schema/financial/poheaddef.c>
#include <schema/financial/polinedef.c>


3
/* -----
           Global variables
----- */

char        errmsg[255];       /* Indicates an error message */
char        po_type[3];        /* Indicates the po type */


4
/* -----
    Screen initialization.
----- */
SCREEN      *pohead_scr=(SCREEN*)NULL;
SCREEN      *poline_scr=(SCREEN*)NULL;
SCREEN      *posel_scr=(SCREEN*)NULL;


5
struct pohead_type   pohead_rec;
struct pohead_type   prev_pohead_rec;

6

/* -----
    Parameter Definition Table and size variable
----- */
struct param_type param_array[] =
    {
    { 'T', po_type, PRM_CHAR, sizeof(po_type)-1, NULL, "po type",
        "Type of PO."},
    { 'o', printer_name, PRM_CHAR, sizeof(printer_name)-1, NULL,
        "printer name", "Purchase order printer."},
    { 's', subsidiary, PRM_CHAR, sizeof(subsidiary)-1, NULL, "subsidiary",
   "Subsidiary type for purchase order"},
    { 'p', perm, PRM_CHAR, sizeof(perm)-1, NULL, "permission",
   "Purchase order permissions"},
    };

int max_params = (sizeof(param_array) / (sizeof(param_array[0])));
```

Legend for the def.c file:

1. mac.h include
2. schema file def.c’s
3. global program variables
4. pointers
5. structure definitions
6. parameters

mac.h Files

Every program uses a macro header (mac.h) file. The mac.h file for a screen-oriented program can contain the following information:

- Includes related to system header files
- Includes related to CX library and other application processes
- Includes for schema files' mac.h files
- Program constant definitions (i.e., *#define* statements)
- Structure definitions

Example of a mac.h File

The following is an edited excerpt from the mac.h file for the Purchasing and Accounts Payable program *purchase*. It appears here to illustrate the common components of a standard CX mac.h file.

```
 1

#include <stdio.h>
#include <string.h>
#include <util/cars.h>

 2

#include <schema/financial/poheadmac.h>
#include <schema/financial/polinemac.h>
#include <schema/financial/bpoheadmac.h>

 3

#define RPA_JRNL_DEBUG
#define PURCH_RUN_CODE "PURCH"
#define ID_REC2       "id_rec2"
#define ID_REC3       "id_rec3"

 4

struct review_type {
    char select_flag;
    struct poline_type poline_rec;
};

struct selreq_type {
    char select_flag;
    struct appritem_type appritem_rec;
    float tcost;
};
```

Legend for the mac.h file:

1. includes for header files
2. includes for schema files
3. program constant definitions
4. structure definitions

SECTION 9 - ACCOUNTS PAYABLE

Overview

Introduction

The *acctspay* program is used in the Accounts Payable application to match purchase order items to received items and to match an invoice to the listed items. It then schedules the payment for the received items and, if applicable, creates a Fixed Asset record.

Program Features Detailed

This section contains details about the following features of the *acctspay* program:

- Process flow
- Parameters
- Program screens

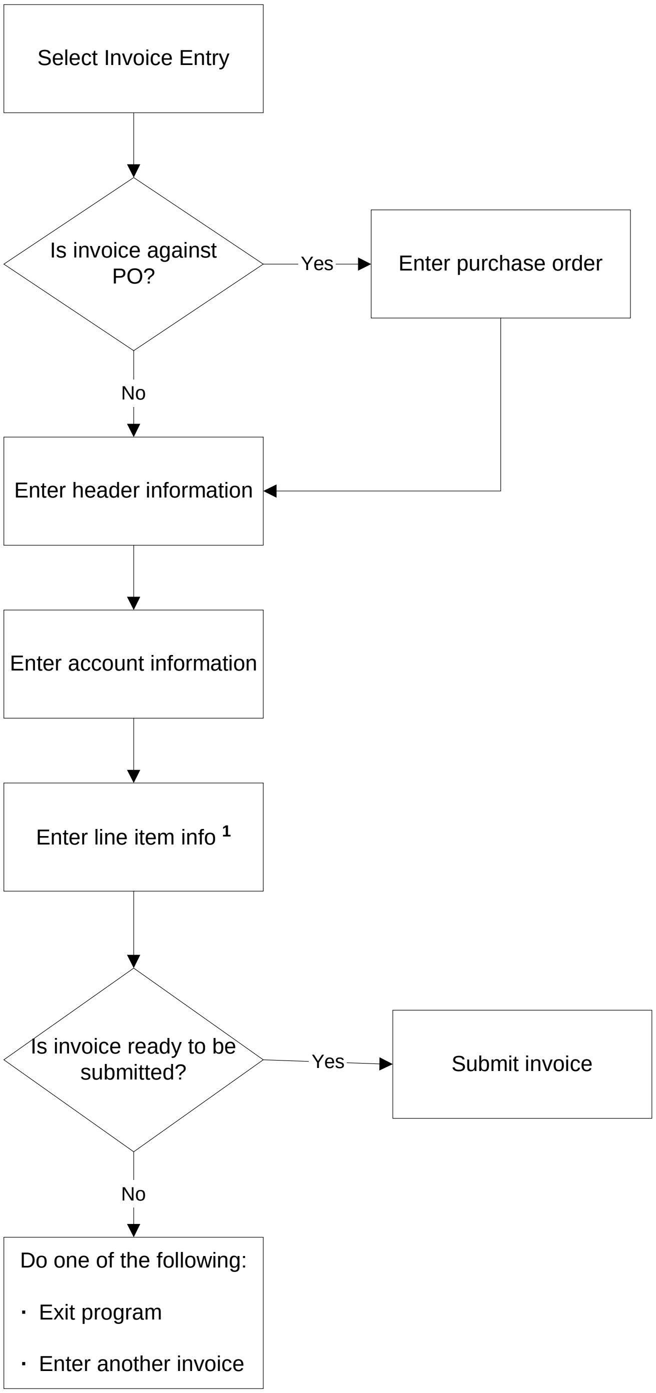
Program Files

All program files for acctspay appear in the following directory:
$CARSPATH/src/purchase/acctspay

Process Flow

Diagram

The following diagram shows the flow of data in the *acctspay* program.



[1] Not required on Simplified Invoice

Process Flow Description

The following describes the process flow in the *acctspay* program.

1. The user accesses Invoice Entry, the menu name for Accounts Payable.
2. The user determines the type of invoice to be processed and selects the appropriate command:
   - Add Invoice—used when invoice relates to a purchase order and line item information must be recorded
   - Add Direct Invoice—used when the invoice does not relate to a purchase order, but line item information must be recorded
   - Fast Invoice—used to create a Simplified Invoice that does or does not relate to a purchase order, but where only the account number and total charge for that account are required. No line item information is recorded.
3. If the invoice relates to a purchase order, the user selects the purchase order. If the invoice does not relate to a purchase order, the user enters the following information:
   - Header
   - Account
   - Line item
4. The user submits the invoice, exits the program, or enters additional invoices.

Program Relationships

The *acctspay* program uses the following records as input.

approve

- appritem_rec

purchase

- poacct_rec
- pohead_rec
- poline_rec

The *acctspay* program creates the following records as input for the General Ledger product.

- gltr_rec
- subb_rec
- sube_rec
- subtr_rec

The *acctspay* program also updates the pendreq_rec.

Accounts Payable Parameters

Introduction

CX contains parameters and compilation values for executing the *acctspay* program. You can specify parameters to compile *acctspay* in a specified manner at the time of execution.

Parameter Syntax

You can display *acctspay* parameters by entering the following: **acctspay -,**

The following is the correct usage for running the *acctspay* program from the UNIX shell:

```
Usage: acctspay [-T AP type] [-o printer name] [-s subsidiary]
          [-n station number]
```

Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required.

Parameters

The following lists the parameters for running *acctspay*.

-T AP type (optional)
Specifies the type of Accounts Payable

Example: acctspay -T IV

-o printer name (optional)
Specifies the name of the printer that you use to print in the Accounts Payable area.

Example: acctspay -o laser

-s subsidiary (optional)
Specifies the type of subsidiary you use with Accounts Payable.

Example: acctspay -s A/P

-n station number (optional)
Specifies the number of the station for document control.

Example: acctspay -n 1

Program Screens

Introduction

The *acctspay* program uses fourteen screens to capture and display accounts payable information.

Access

The screen files for *acctspay* are located in the following directory path:
$CARSPATH/modules/purchase/progscr/acctspay

Screen Files and Table/Record Usage

The *acctspay* screens appear in the following files and use the indicated tables and records:

account
Contains the Line Item Account Entry screen.
Tables/Records: invacct_rec, subb_table, subt_table

apparm
Contains the Accounts Payable - Parameter screen.
Tables/Records: form_table, sortreq_table, subs_table

apselreq
Contains the Accounts Payable - Requisition Selection screen.
Tables/Records: appritem_rec, fscl_cal_rec

credacct
Contains the Line Item Account Entry screen.
Tables/Records: invacct_rec, subb_table, subt_table

credlist
Contains the Listing of Queried Credit Memos screen.
Tables/Records: id_rec, invhead_rec

credmemo
Contains the Accounts Payable Credit Memo screen.
Tables/Records: f1099_table, form_table, id_rec, invhead_rec

fastrev
Contains the Accounts Payable Review screen used to process simplified invoices.
Tables/Records: id_rec, invacct_rec, invhead_rec

invhead
Contains the Accounts Payable Direct Entry screen.
Tables/Records: f1099_table, form_table, id_rec, invhead_rec, pay_term_table

invline
Contains the Accounts Payable Line Item Entry screen.
Tables/Records: invacct_rec, invline_rec, subb_table, subt_table

invlist
Contains the Listing of Queried Invoices screen.
Tables/Records: id_rec, invhead_rec

invrev
Contains the Accounts Payable Review screen.
Tables/Records: f1099_table, form_table, id_rec, invhead_rec, invline_rec, pay_term_table

manualap

Contains the Form A/P # parameter screen.
Tables/Records: none

payee

Contains the Accounts Payable Payee Entry screen.
Tables/Records: id_rec, invhead_rec

polook

Contains the Purchase Order Listing screen.
Tables/Records: id_rec, pohead_rec

SECTION 10 - APPROVAL

Overview

Introduction

This section provides reference information about the Approval (*approval*) program. The *approval* program enables users to review requisitions, purchase orders and invoices in order approve or deny the purchase of requisitioned items. This process can automatically process approvals in a vertical and/or horizontal method, or a combination of the two. Approvals can also occur at the ID, commodity code, and/or account levels.

Vertical (Serial) Processing

Vertical processing means approval is based on receiving one person's approval before a second person can approve it. When the institution's authorization tables contain multiple authorizing ID numbers with different dollar limits, the approval program uses the vertical processing method.

Horizontal (Parallel) Processing

Horizontal processing means that approvals can be made by several people simultaneously. All approvers must approve the document before it can pass to the next approval step. When the institution's authorization tables contain multiple authorizing ID numbers with the same dollar limits, the *approval* program uses the horizontal processing method.

Approval at the ID Level

By using the auth_id_table, an institution can enable an approver to approve all the specified documents for one or more specific individuals. For this type of approval, the *approval* program matches the ID number of the user who submits the document with the id field in the auth_id_table. It then locates the auth_id on the table entry, and associates the document with the auth_id. In addition, the program compares the prices on the document to the dollar_limit field in the auth_id_table. Only those documents within the dollar limit are routed to the approver.

Approval at the Account Level

By using the auth_acct_table, an institution can enable an approver to approve all the documents that charge goods or services to a specific account or range of accounts. For this type of approval, the *approval* program matches the account charged with the ranges of account components (fund, function, object and subfund) in the auth_acct_table. In addition, the program compares the prices on the document to the dollar_limit field in the auth_acct_table. Only those documents within the dollar limit are routed to the approver.

Approval at the Commodity Code Level

By using the auth_comm_table, an institution can enable an approver to approve all the documents for one or more specific commodity codes. For this type of approval, the approval program matches the commodity code on the document to the comm_code field in the auth_comm_table. In addition, the program compares the prices on the document to the dollar_limit field in the auth_comm_table. Only those documents within the dollar limit are routed to the approver. .

Program Features Detailed

The following aspects of the *approval* program are detailed in this section:

- Process flow
- Program processing
- Program screens

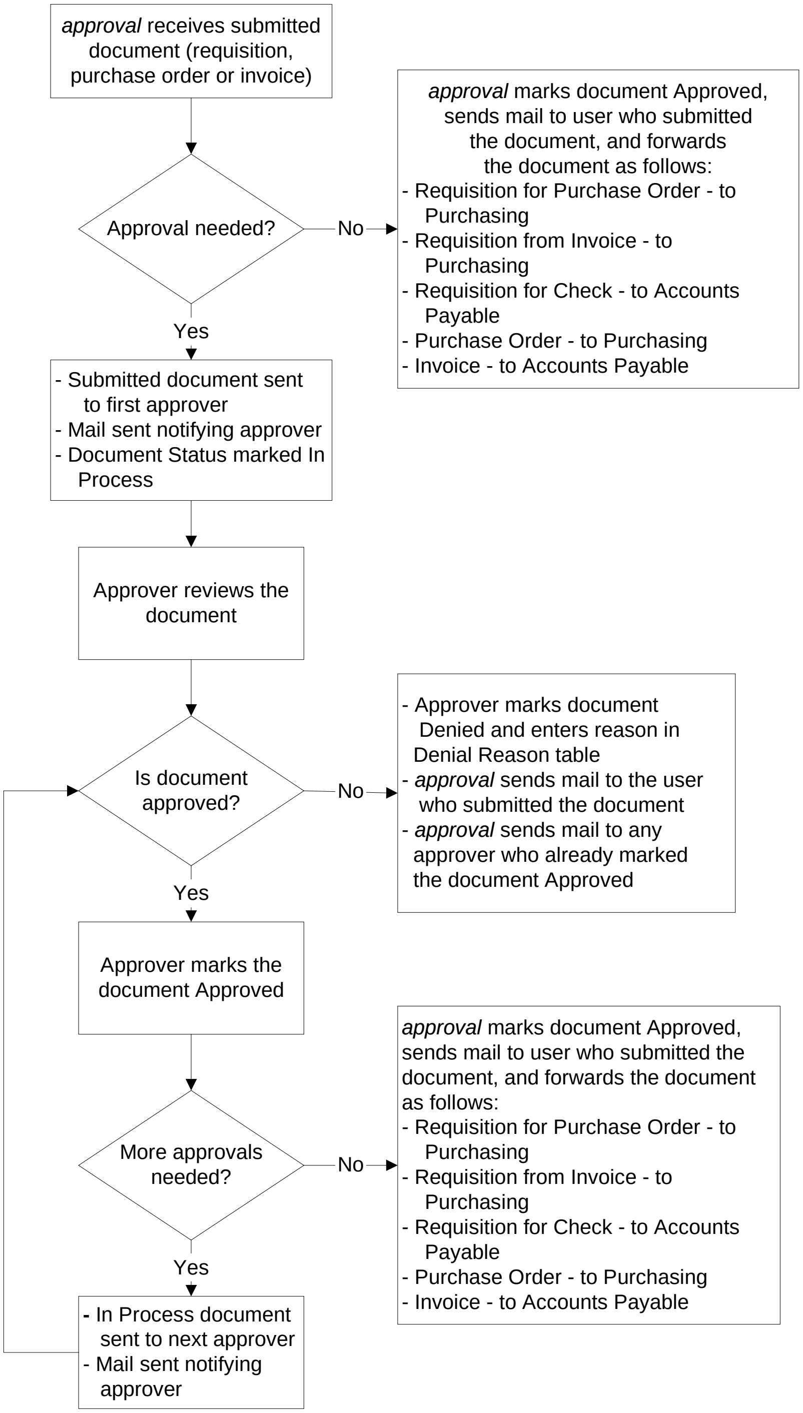
Program Files

All the program files for *approval* are located in the following directory:
$CARSPATH/src/requisition/approval

Process Flow

Diagram

The following diagram shows the flow of data in *approval*.


approval receives submitted document (requisition, purchase order or invoice)
Approval needed?
No
approval marks document Approved, sends mail to user who submitted the document, and forwards the document as follows:
- Requisition for Purchase Order - to Purchasing
- Requisition from Invoice - to Purchasing
- Requisition for Check - to Accounts Payable
- Purchase Order - to Purchasing
- Invoice - to Accounts Payable
Yes
- Submitted document sent to first approver
- Mail sent notifying approver
- Document Status marked In Process
Approver reviews the document
Is document approved?
No
- Approver marks document Denied and enters reason in Denial Reason table
- approval sends mail to the user who submitted the document
- approval sends mail to any approver who already marked the document Approved
Yes
Approver marks the document Approved
More approvals needed?
No
approval marks document Approved, sends mail to user who submitted the document, and forwards the document as follows:
- Requisition for Purchase Order - to Purchasing
- Requisition from Invoice - to Purchasing
- Requisition for Check - to Accounts Payable
- Purchase Order - to Purchasing
- Invoice - to Accounts Payable
Yes
- In Process document sent to next approver
- Mail sent notifying approver

Program Processing

Status of Approval Records

The *approval* program adds an Approval record for each required approval. These records provide a way to track the document (requisition, purchase order or invoice) as it processes through the system.

When a document is waiting for approval, both the document and the Approval record have a status of S (Submitted for Approval). When a document is partially approved, both the document and the Approval record have a status of I (In progress). When a document receives final approval, the status of both the document and the Approval record changes to A (Approved). If a document is denied at any point in the process, the status of both the document and the Approval record changes to D (Deny).

When a user needs to stop the approval process for a document with a status of I (In Progress), the Hold command removes a requisition from the approval process, and the Forget Approval command removes a Purchase Order or an Invoice from the approval process. When a document is removed from the approval process, the approval record(s) for that document no longer exists, and the document status is changed to H (Hold) on a requisition and to N (New) on a purchase order or an invoice.

Approval Process for Approved Documents

The Approval process begins when a submitted document is added to the approval record. In the first phase of the approval process, the *approval* program goes through the following steps:

1. Determines who must approve the document by selecting records from the auth_id_table, auth_acct_table, and auth_comm_table.
2. Sorts approvers by authorization limit and compares to the document cost.
3. Notifies the first approver by mail that a document needs approval, and adds all approv_recs for an audit trail.
4. Determines from the approv_rec who must provide the next approval for the document (if one or more approvals are required). If none are required, it sets the status of the document to A (Approved). If another approval is required, the *approval* program sets the status of the document to I (In progress).
5. Notifies the user who submitted the document that the first approval is complete.
6. Continues the procedures above until all approvals have been received. Then, notifies the user who submitted the document that approval is complete, sets the status of the document to A (Approved), and routes/forwards the document to the appropriate program (see diagram on previous page).

Approval Process for Denied Documents

When the approver denies a document, *approval* completes the following steps:

1. Verifies that the denial code is a valid entry in the Denial table.
2. Sets the status of the document in the header record (reqhead_rec, pohead_rec, invhead_rec) to D (Denied).
3. Updates the approv_rec to remove all remaining entries for the denied document. Therefore, additional approvers do not see the denied document in their list of documents waiting to be approved.
4. Notifies the user who submitted the document and all approvers who approved it, through electronic mail, that the document was denied. The mail message includes the denial reason.
5. Sets the status of the approv_rec to D (Denied), indicating that the approver denied the document.
6. Adds a denial_rec indicating the document and the reason for the denial of the document.
7. Adjusts the pending requisition amounts from the pendreq_rec, if the document is a requisition.

Program Screens

Introduction

The *approval* program uses four screens to display documents and to enable approvers to approve or deny documents.

Access

The screen files for *approval* are located in the following directory path:
$CARSPATH/modules/requisition/progscr/approval

Screen Files and Table/Record Usage

The *approval* screens appear in the following files and use the indicated tables and records.

aprvacct
Contains the Account Review screen
Table/Records: reqhead_rec, pohead_rec, invhead_rec

aprvsel
Contains the Approval Selection screen
Table/Records: reqhead_rec, pohead_rec, invhead_rec, id_rec, userid_table

aprvsumm
Contains the Requisitions for Approval screen
Table/Records: reqhead_rec, reqline_rec, pohead_rec, poline_rec, invhead_rec, invline_rec

denial
Contains the Requisition Denial screen
Table/Records: denial_rec, denial_table, denial_blob

params

Contains the Approval Parameter Entry screen

Table/Records: none

SECTION 11 - ASSIGN BUYER

Overview

Introduction

The Assign Buyer program (*assign)* enables one or more users to assign approved requisitions to a specific station number or buyer, where they can be used to generate purchase orders.

Program Features Detailed

This section contains details about the following features of the *assign* program:

- Process flow
- Parameters
- Program screens

Program Files

All the program files for *assign* appear in the following directory:
$CARSPATH/src/purchase/assign

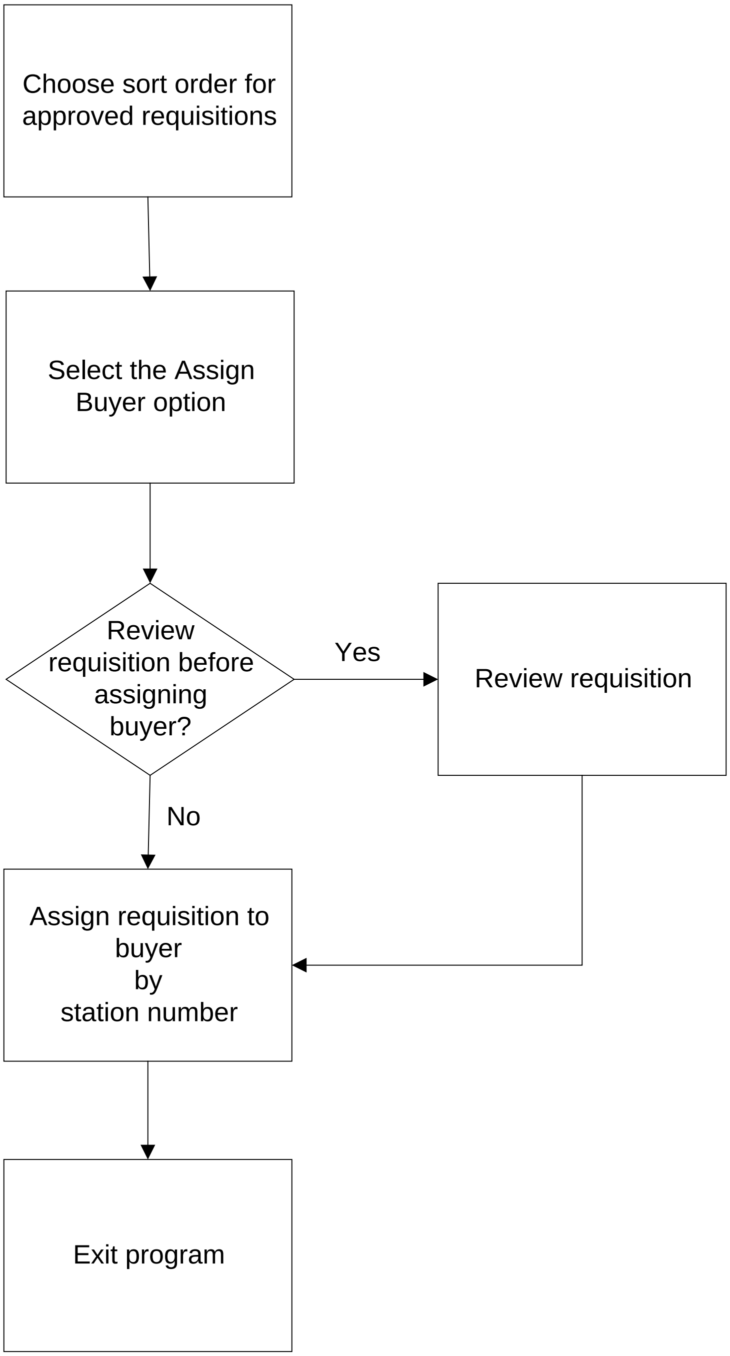
Program Access

Users can access *assign* from the Fiscal Management: PO/Requisition/Approval menu.

Process Flow

Diagram

The following diagram shows the flow of data in the *assign* process.



Choose sort order for approved requisitions
Select the Assign Buyer option
Review requisition before assigning buyer?
Yes
Review requisition
No
Assign requisition to buyer by station number
Exit program

Process Flow Description

The following process describes the process flow in the *assign* program.

1. The user selects the sort order by which to view the approved requisitions.
2. The user selects the **Assign** command to display the list of approved requisitions, assigned and unassigned, which have not yet been added to a purchase order and submitted.
3. If desired, the user reviews the requisitions in detail before assigning it to a buyer.
4. The user enters the station number of the buyer, and the database immediately reflects the assignment.
   Note: Table Lookup enables the user to view a list of the stations and the buyer associated with each.
5. The user exits from the program.

Program Relationships

The *assign* program uses the requisitions from the *approval* program. It works with the *purchase* program by limiting the requisitions a buyer can access.

Program Screens

Introduction

The *assign* program uses two screens: one parameter screen, and one selection screen.

Access

The screen files are located in the following directory path:
$CARSPATH/modules/purchase/progscr/assign

Screen Files

The *assign* screens appear in the following files:

abinit
Contains the Requisition Sort Order Parameter screen.
Tables/Records: sortreq_table

abselreq
Contains the Assign Requisition to Buyer screen.
Tables/Records: appritem_rec, buyer_table, reqhead_rec

SECTION 12 - MASS INVOICE

Overview

Introduction

The *massinv* program is used in the Mass Invoice application to enable users to enter invoice information without using requisitions or purchase orders. The purpose of this process is to create actual charges against accounts without using the *requisition* or *purchase* programs, therefore minimizing data entry.

The *massinv* program is an optional, stand-alone process that does not interact with any of the Requisitioning, Purchasing, and Accounts Payable applications. Invoices created in the Mass Invoice application cannot be accessed via the *acctspay* program used for Invoice Entry in the Requisitioning, Purchasing, and Accounts Payable product.

Program Features Detailed

This section contains details about the following features of the *massinv* program:

- Process flow
- Parameters
- Program screens

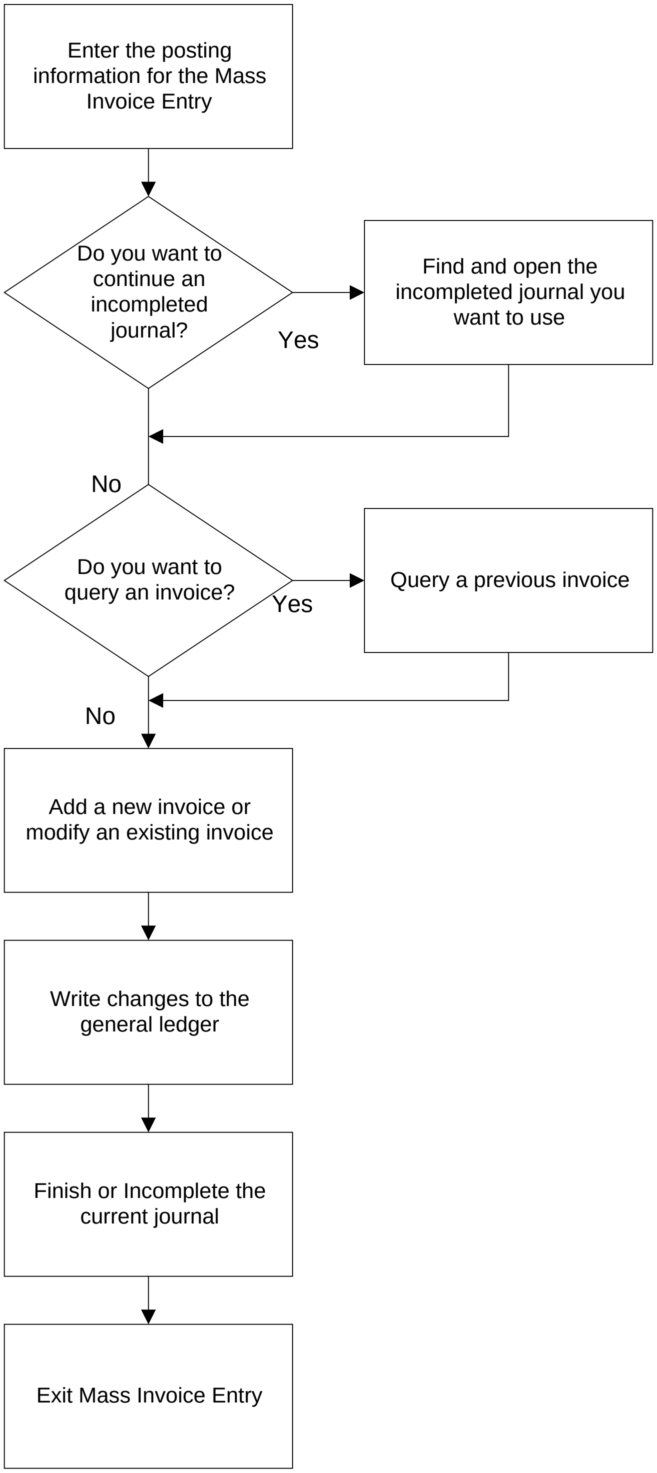
Program Files

All program files for *massinv* appear in the following directory: $CARSPATH/src/purchase/invoice

Process Flow

Diagram

The following diagram shows the process flow in the *massinv* program.


Enter the posting information for the Mass Invoice Entry
Do you want to continue an incompleted journal?
Yes
Find and open the incompleted journal you want to use
No
Do you want to query an invoice?
Yes
Query a previous invoice
No
Add a new invoice or modify an existing invoice
Write changes to the general ledger
Finish or Incomplete the current journal
Exit Mass Invoice Entry

Data Flow Description

The following process describes the data flow in the *massinv* program.

1. The user accesses Mass Invoice Entry.
2. The user enters the posting information for the invoices.
3. Does the user want to continue an incompleted journal?
   - If yes, the user selects the **Journal** command to locate the desired journal, then goes to step 4.
   - If no, the user goes to step 4.
4. Does the user want to query an invoice?
   - If yes, the user performs the query, then goes to step 5.
   - If no, the user goes to step 5.
5. The user adds a new invoice or changes the existing invoice.
6. The user writes the changes to the general ledger.
7. The user either finishes the journal, or marks it Incomplete.
8. The user exits Mass Invoice Entry.

Program Relationships

The *massinv* program provides input to the *ckslct* program.

Parameters

Introduction

CX contains parameters or compilation values for executing the *massinv* program You can specify parameters to compile *massinv* in a specified manner at the time of execution.

Parameter Syntax

You can display *massinv* parameters by entering the following: **invoice -,**

The following is the correct usage for running the invoice program from the UNIX shell:

```
Usage: invoice [-a runcode] [-c doc_code] [-d rundate] [-f fund] [-s subs] [-z]
```

Note: Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required.

Parameters

The following list contains the parameters for running *massinv*.

-a runcode (optional)

Specifies the ADR runcode associated with the invoices. This runcode enables users to select the type of address to which they want payment to be sent (e.g., PERM, INVOICE, SINGLE).

Example: invoice -a INVOICE

-c doc_code (optional)

Specifies the document code to use with the invoices.

Example: invoice -c IV

-d rundate (optional)

Specifies the posting date to use for the invoices, in the format mm/dd/yyyy.

Example: invoice -d 01/02/1997

-f fund (optional)

Specifies the default fund to use for all the invoice transactions. A value entered in this field overrides the default defined in the macro DEFAULT_FUND.

Example: invoice -f 10

-s subs (optional)

Specifies the subsidiary to use for the invoice transactions.

Example: invoice -s A/P

-z (optional)

Indicates that you want to process the *massinv* program in debug mode.

Example: invoice -z

Program Screens

Introduction

The *massinv* program uses five screens to capture and display mass invoice information.

Access

The screen files for *massinv* are located in the following directory path:
$CARSPATH/modules/accounting/progscr/invoice

Screen Files and Table/Record Usage

The *massinv* screens appear in the following files and use the indicated tables and records:

continue
Contains the Mass Invoice Entry - Incomplete Journals window.
Tables/Records: vch_rec

invmode
Contains the Mass Invoice Entry screen.
Tables/Records: f1099_table, invchg_rec, invctgry_table, invoice_rec, invsubs_rec, pay_term_table, suba_rec, subb_rec, subs_table

param
Contains the Mass Invoice Parameter screen.
Tables/Records: adr_rec, doc_table, subs_table

query
Contains the Mass Invoice Entry - Query Results screen.
Tables/Records: gle_rec, subb_rec, sube_rec

subs
Contains the Mass Invoice Entry - Subsidiary window.
Tables/Records: invsubs_rec, subb_table, subt_table

SECTION 13 - PURCHASE

Overview

Introduction

The *purchase* program is used to combine requisition information to generate a purchase order, or to create purchase orders when a user directly enters information. After the user enters data and saves it to the database, the program assigns a number that tracks the purchase order through the system.

When the user enters the **Submit** command, the program updates the purchase order status to *Submitted*, indicating the purchase order is ready to be issued to the vendor and the cost of the items is encumbered in the general ledger accounts. The program also prints the purchase order automatically.

The *purchase* program is also used to issue change orders to vendors.

Program Features Detailed

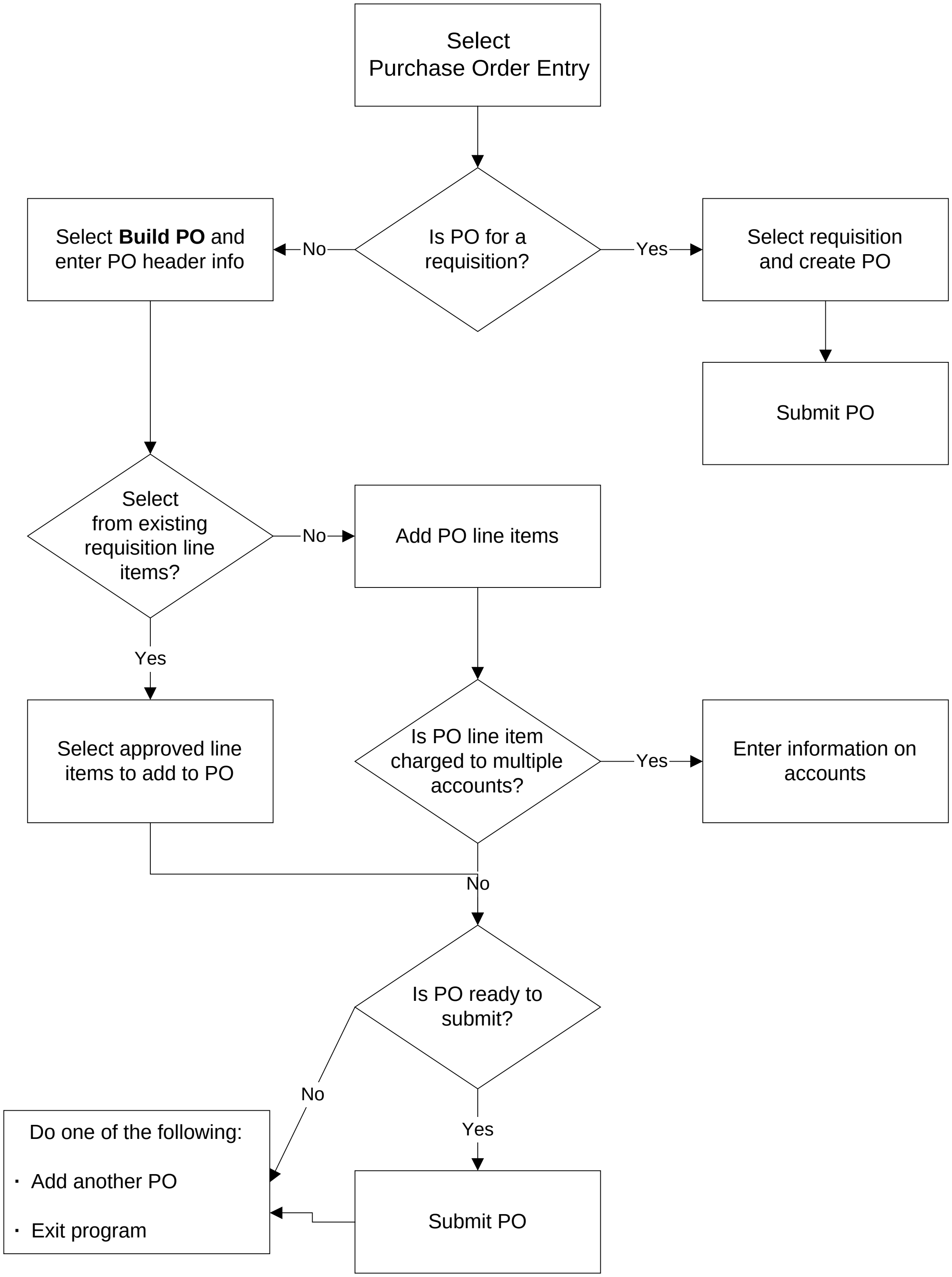
This section contains details about the following features of the *purchase* program:

- Process flow
- Parameters
- Program screens

Process Flow

Diagram

The following diagram shows the process flow in the *purchase* program.


Select Purchase Order Entry
Is PO for a requisition?
No
Select **Build PO** and enter PO header info
Yes
Select requisition and create PO
Submit PO
Select from existing requisition line items?
No
Add PO line items
Yes
Select approved line items to add to PO
Is PO line item charged to multiple accounts?
Yes
Enter information on accounts
No
Is PO ready to submit?
No
Yes
Do one of the following:
· Add another PO
· Exit program
Submit PO

Data Flow Description

The following process describes the data flow in the *purchase* program.

1. The user accesses the *purchase* program.
2. Is the purchase order for a requisition?
   - If yes, the user selects the requisition, creates and submits a purchase order, and then does one of the following:
     - Processes other purchase orders
     - Exits from the program
   - If no, the user builds a purchase order.
3. Does the purchase order information come from existing requisition line items?
   - If yes, the user selects items to include on the purchase order.
   - If no, the user directly enters purchase order line items.
4. The user enters account information.
5. Does the user want to submit the purchase order?
   - If yes, the user selects the **Submit** command and the purchase order encumbers the account(s) charged.
   - If no, the user selects the Close command and the purchase order information is stored for later submission.

Program Relationships

The *purchase* program can use information from approved requisitions. The requisitions originate in *requisition*, and obtain approval in *approve*.

In addition, the purchase orders created in the *purchase* program use the General Ledger program *bgvoucher* to post encumbrances to a journal and to the General Ledger. The encumbrances also are input to *acctspay*.

Purchase Parameters

Introduction

CX contains parameters and compilation values for executing the *purchase* program. You can specify parameters to compile *purchase* in a specified manner at the time of execution.

Parameter Syntax

You can display *purchase* program parameters by entering the following: **purchase -,**

The following is the correct usage for running the *purchase* program from the UNIX shell:

```
Usage: purchase [-T po type] [-o printer name] [-s subsidiary]
```

Note: Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required. If you do not enter the optional parameters at the time you execute the program from the UNIX shell, the program automatically displays a parameter screen with the parameters and their defaults.

Parameters

The following lists the parameters for running *purchase*.

-T po type (optional)
Specifies the type of purchase order.

Example: purchase -T PO (for an open purchase order type)

-o printer name (optional)
Specifies the name of the printer the institution uses for purchase orders.

Example: purchase -o laser

-s subsidiary (optional)
Specifies the name of the subsidiary used with purchase orders

Example: purchase -s A/P

Program Screens

Introduction

The *purchase* program uses eleven screens to capture and display purchase order information.

Access

The screen files for *purchase* are located in the following directory path:
$CARSPATH/modules/purchase/progscr/purchase

Screen Files and Table/Record Usage

The *purchase* screens appear in the following files and use the indicated tables and records:

aalook
Contains the Alternate Address Listing screen.
Tables/Records: aa_rec

cklist
Contains the Listing of Invoices and Checks For This PO screen.
Tables/Records: invhead_rec

commlist
Contains the Commodity Code Listing screen.
Tables/Records: commc_table

denial
Contains the Requisition Denial screen.
Tables/Records: denial_blob, denial_table, linedenial_rec

manualpo
Contains the user specified purchase order number screen. Use this screen when station number allows users to specify the document number.
Tables/Records: form_table, pohead_rec

pohead
Contains the Purchase Order Header Entry screen.
Tables/Records: form_table, id_rec, pohead_blob, pohead_rec

poinit
Contains the Purchase - Parameter screen.
Tables/Records: form_table, sortreq_table, subs_table

poline
Contains the Purchase Order Line Item Entry screen.
Tables/Records: bldg_table, facil_table, id_rec, poacct_rec, pohead_rec, poline_blob, poline_rec, subb_table, subt_table

polineact
Contains the Line Item Account Entry screen.
Tables/Records: poacct_rec, poline_rec, subb_table, subt_table

polist
Contains the Listing of Queried Purchase Orders screen.
Tables/Records: id_rec, pohead_rec

poreview

Contains the Purchase Order Review screen.
Tables/Records: id_rec, pohead_rec, poline_rec

posel

Contains the Purchasing - Requisition Line Item Selection screen.
Tables/Records: appritem_rec, fscl_cal_rec

vendor

Contains the Suggested Vendor Information screen.
Tables/Records: id_rec, pohead_rec, st_table

poselreq

Contains the Purchasing - Requisition Selection screen.
Tables/Records: appritem_rec, fscl_cal_rec

SECTION 14 - PURCHASE ORDER/INVOICE AUDIT

Overview

Introduction

This section provides you with the reference information about the PO/Invoice Audit (*poinvaudit*) program. The poinvaudit program enables users to maintain accounts payable records and balances throughout their institution. This program automatically checks records, and when run with update (*poinvaudit.u*), it automatically updates some records.

Program Features Detailed

This section details the following aspects of the *poinvaudit* program:

- Process flow
- Program processing
- Program parameters

Program Files

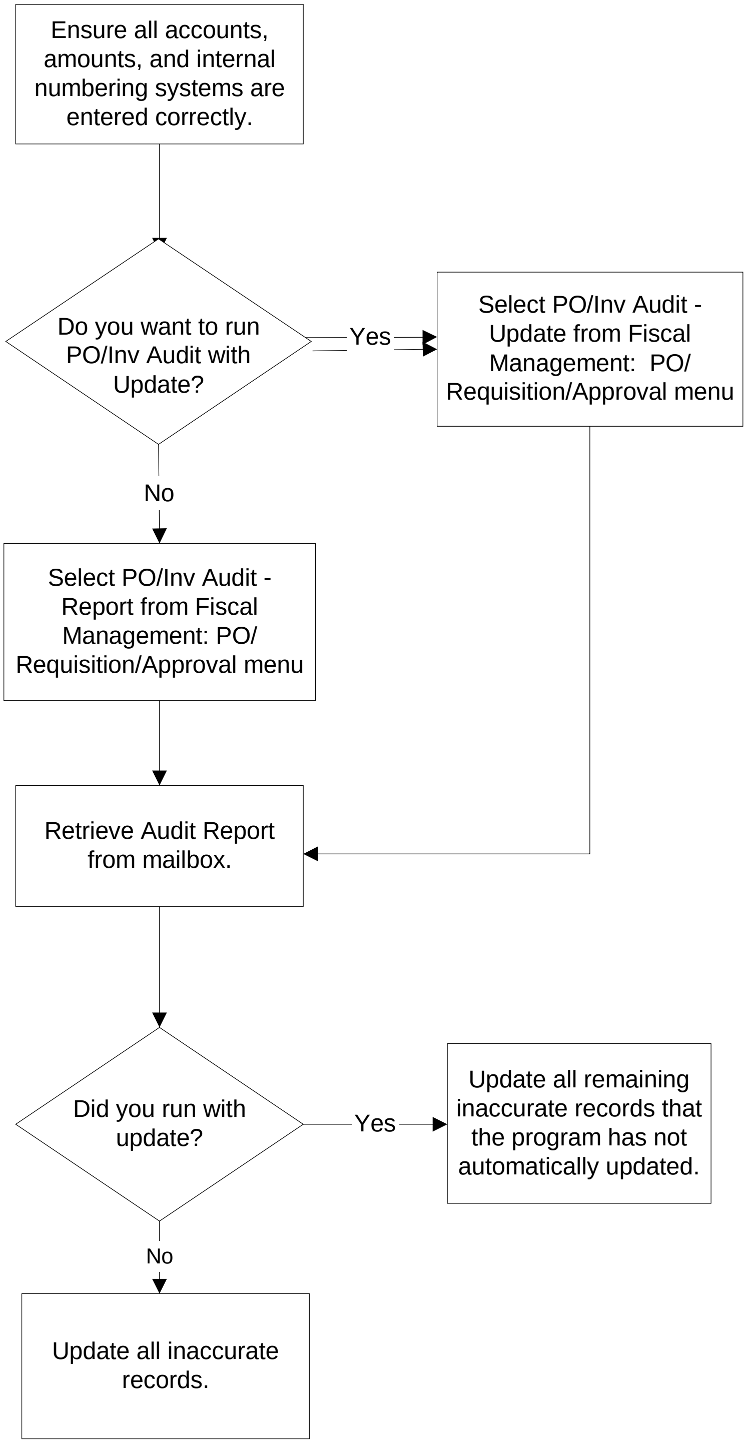
All the program files for poinvaudit are located in the following directory:

$CARSPATH/src/purchase/poinvaudit

PO/Invoice Audit Process Flow

Diagram

The following diagram shows the flow of data in *poinvaudit*.


Ensure all accounts, amounts, and internal numbering systems are entered correctly.
Do you want to run PO/Inv Audit with Update?
Yes
Select PO/Inv Audit - Update from Fiscal Management: PO/ Requisition/Approval menu
No
Select PO/Inv Audit - Report from Fiscal Management: PO/ Requisition/Approval menu
Retrieve Audit Report from mailbox.
Did you run with update?
Yes
Update all remaining inaccurate records that the program has not automatically updated.
No
Update all inaccurate records.

Process Flow Description

The following describes the process flow in *poinvaudit*.

Note: Before running this audit make sure that no journals are open, and all users are out of the Accounts Payable and Purchasing applications.

1. Ensure that all accounts, amounts, and internal numbering systems have been correctly entered.
2. Do you want to run PO/Invoice Audit- Update.
   - If yes, select **PO/Invoice Audit - Update**.
   - If no, select **PO/Invoice Audit Report**.
3. Retrieve the report from your mailbox.
4. Did you run **PO/Invoice Audit- Update**?
   - If yes, then go to step 6.
   - If no, then go to step 5.
5. Update all records that the report has found inaccurate.
6. Update all records that the report has found inaccurate, and that the system has not automatically updated.

Program Processing

Introduction

The *poinvaudit* program enables you to validate the integrity of your accounts payable records and balances, and automatically update some inaccuracies.

PO/Invoice Audit Process

You access the PO/Invoice Audit application by selecting the menu option from the Fiscal Management: PO/Requisition/Approval menu.

You may run the *poinvaudit* and process at any time. Jenzabar recommends that you run it frequently to ensure the integrity of your data. After entering purchase order and invoice information you may run *poinvaudit.*

When you run the process the *poinvaudit* program runs a series of 23 checks to verify data integrity. For more information about each of these checks, refer to the CXguide *Using Purchasing and Accounts Payable*.

SECTION 15 - RECEIVING

Overview

Introduction

The *receive* program is used in the Receiving application to process information about the receipt of items requested on a purchase order. When you receive a shipment from a vendor, use the *receive* program to do the following tasks:

- Select the invoice line items matching the received items
- Record the quantity of items received
- Record whether the items are fully or partially received, back ordered or unavailable
- Record the condition of received items

The *receive* program is also used to record the return of received items. When you return items to a vendor, use the *receive* program to do the following tasks:

- Select the received items on the receiving record
- Record the quantity of items being returned
- Record the reason for returning the items
- Record the method of shipping returned items
- Record approved code for the return

Program Features Detailed

The following aspects of the *receive* program are detailed in this section:

- Process flow
- Program screens

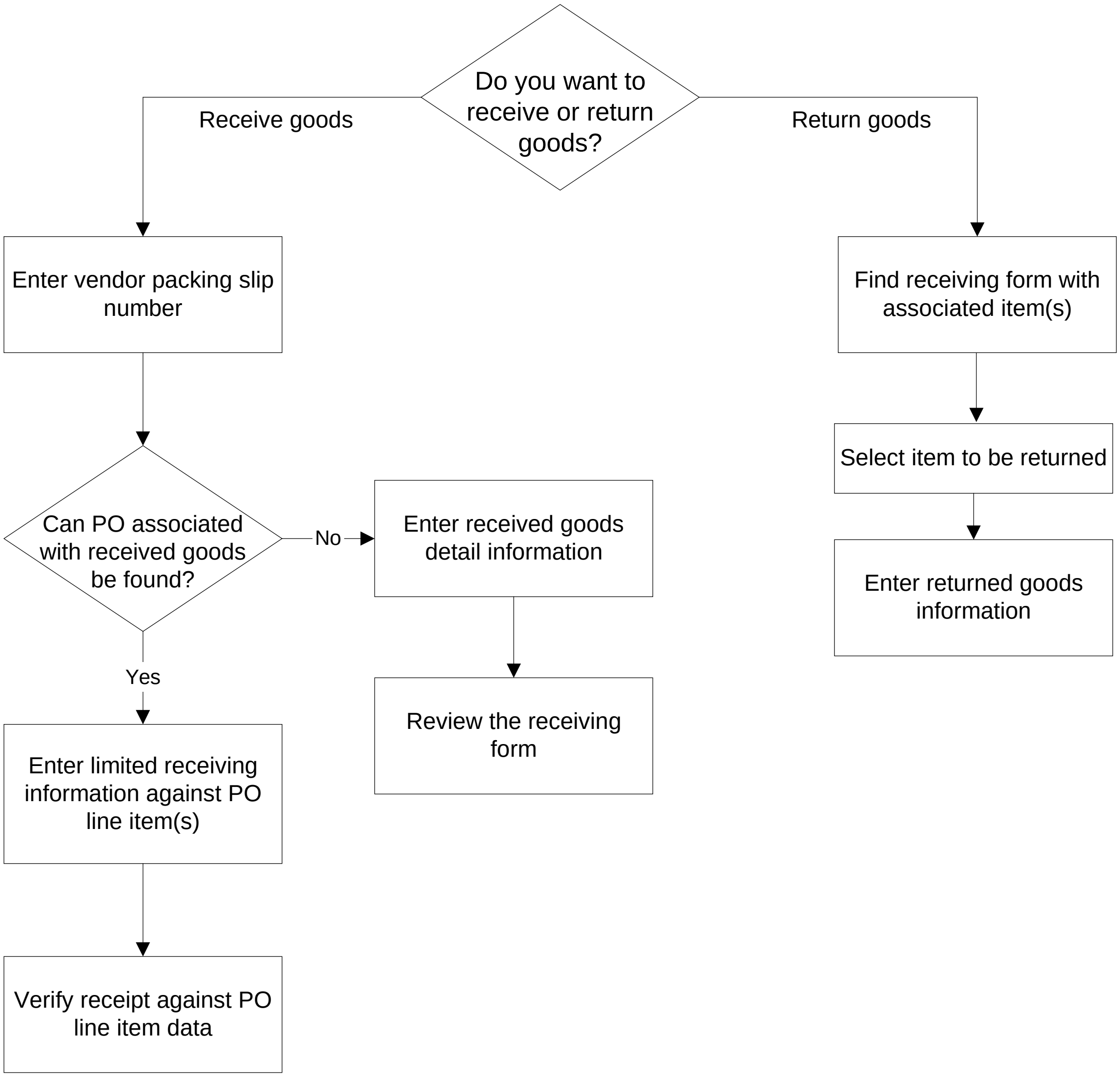
Parameters

The *receive* program does not require parameters or compilation values for execution.

Process Flow

Diagram

The following diagram shows the flow of data in the *receive* program. .



Do you want to receive or return goods?
Receive goods
Return goods
Enter vendor packing slip number
Find receiving form with associated item(s)
Select item to be returned
Enter returned goods information
Can PO associated with received goods be found?
No
Enter received goods detail information
Review the receiving form
Yes
Enter limited receiving information against PO line item(s)
Verify receipt against PO line item data

Data Flow Description

The following process describes the data flow in the *receive* program.

1. The user accesses Receiving Entry.
2. Does the user want to enter receiving or return information?
   - If <u>receiving</u>, go to step 3.
   - If <u>return</u>, go to step 6.
3. The user enters a vendor packing slip number.
4. Can the user locate the purchase order associated with the received goods?
   - If <u>yes</u>, the user enters receiving information against the purchase order line items, and verifies the receipt against the purchase order line item information.
   - If <u>no</u>, the user enters detail information about the received goods, and reviews the receiving form.
5. Does the user need to enter more receiving information?
   - If <u>yes</u>, the user goes to step 2.
   - If <u>no</u>, the user exits the program.
6. The user locates the receiving form with the associated items, selects the item to be returned, and enters returned goods information.
7. Does the user need to enter more receiving information?
   - If <u>yes</u>, the user goes to step 2.
   - If <u>no</u>, the user exits the program.

Program Relationships

The *receive* program accesses purchase orders from *purchase* and provides information to *acctspay*.

Program Screens

Introduction

The *receive* program uses nine screens to capture and display receiving information.

Access

The screen files are located in the following directory path:
$CARSPATH/modules/purchase/progscr/receive

Screen Files and Table/Record Usage

The *receive* screens appear in the following files and use the indicated tables and records:

polook
Contains the Purchase Order Listing screen.
Tables/Records: id_rec*,* pohead_rec

poltrecv
Contains the Review of Received Items Against PO Item screen.
Tables/Records: poline_rec, rechead_rec, recline_rec

rechead
Contains the Receiving Header Entry screen.
Tables/Records: form_table, id_rec, rechead_rec

recinfo
Contains the Receiving **-** Information screen.
Tables/Records: bldg_table, cond_table, facil_table, id_rec*,* rechead_rec, recinfo_rec, recline_rec, units_table

recline
Contains the Receiving **-** Line Item Entry screen.
Tables/Records: bldg_table, cond_table, id_rec, poline_blob, poline_rec*,* rechead_rec, recline_blob, recline_rec, units_table

reclist
Contains the Listing of Queried Receiving Forms screen.
Tables/Records: id_rec, rechead_rec

recsumm
Contains the Receiving **-** PO Summary screen.
Tables/Records: id_rec, poline_rec, rechead_rec

retrev
Contains the Return Goods Review screen.
Tables/Records: id_rec, poline_rec, rechead_rec, return_rec

return
Contains the Return screen.
Tables/Records: id_rec, poline_rec, rechead_rec, recline_rec, return_blob, return_rec, retreason_table, shipmethod_table, units_table

review
Contains the Receiving Review screen.
Tables/Records: id_rec, poline_rec, rechead_rec, recline_rec

SECTION 16 - REQUISITION

Overview

Introduction

This section provides you with reference information about the *requisition* program. The Requisitioning product uses *requisition* to accumulate information about goods or services needed at your institution. Using the information from *requisition*, Purchasing can create a complete purchase order.

Program Features Detailed

The following aspects of the *requisition* program are detailed in this section:

- Process flow
- Program processing
- Program parameters

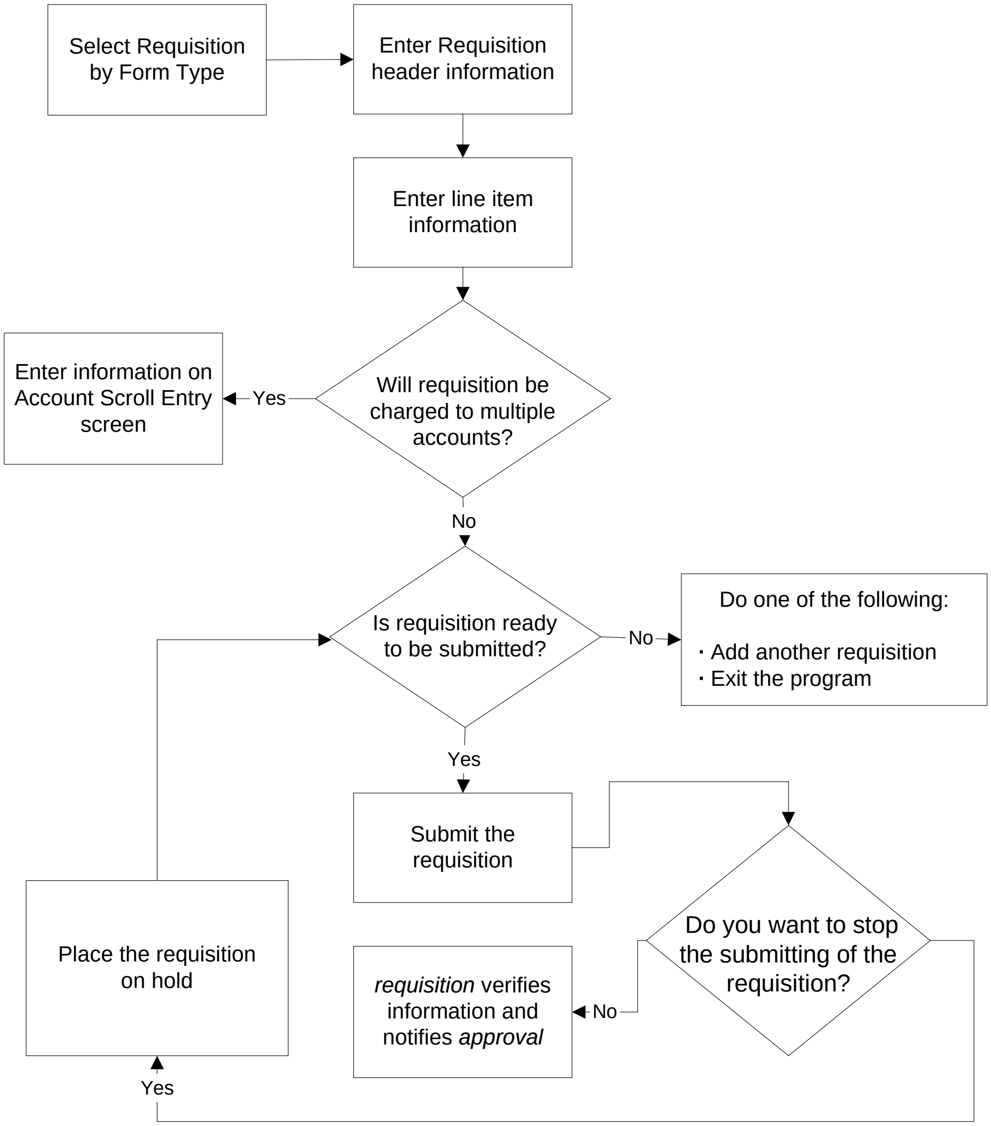
Program Files

All the program files for *requisition* are located in the following directory:
$CARSPATH/src/requisition/requisition

Requisition Process Flow

Diagram

The following diagram shows the flow of data in *requisition*.


Select Requisition by Form Type
Enter Requisition header information
Enter line item information
Will requisition be charged to multiple accounts?
Yes
Enter information on Account Scroll Entry screen
No
Is requisition ready to be submitted?
No
Do one of the following:
· Add another requisition
· Exit the program
Yes
Submit the requisition
Do you want to stop the submitting of the requisition?
No
requisition verifies information and notifies *approval*
Yes
Place the requisition on hold

Process Flow Description

The following describes the process flow in *requisition*.

1. The user enters requisition header information (i.e., information that relates to the entire requisition).
2. The user enters information about the individual requisitioned items, including an account number to which to charge the item.
3. If required, the user accesses the Account Scroll Entry screen to charge the requisitioned items to more than one account.
4. Is the user ready to submit the requisition?
    - If yes, the user selects **Submit**. The program verifies the information on the requisition and notifies *approval* that a submitted requisition requires review and approval from the designated approver.
    - If no, the user adds another requisition or exits the program.
5. Is the user finished adding requisitions?
    - If no, the user repeats steps 1 - 4.
    - If yes, the user selects **Exit**.

Program Relationships

Requisitions added through the requisition process are input to the *approval* program. The user must set up the auth_id_table, the auth_acct_table or the auth_comm_table to run *approval*. If none of these tables are set up, the *requisition* program will mark the requisitions as approved and immediately route them to Purchasing and Accounts Payable.

Program Processing

Introduction

The *requisition* program enables you to enter requisition information about goods and services that your institution orders.

Requisitioning Process

You access the Requisitioning application by selecting the menu option that defines the type of requisition you want to generate. The menu options you can select from are:

- Requisition for Purchase
- Requisition for Invoice
- Requisition for Check

Note: The selection of the requisition type affects the sequence for the numbering of requisitions.

After entering the necessary information for the selected requisition form type, you may submit the requisition to the approval process or leave it for later processing. When the requisition is submitted to the approval process, the *requisition* program's submit logic verifies the following requirements:

- All required fields must contain valid information
- At least one item must be listed on the requisition
- Determines the approvals necessary before the requisition is made available to Purchasing

In addition, as requisition line items are entered and again when they are submitted, the program checks the accounts indicated on the requisition. First, it checks the general ledger permission tables to verify that the user is approved to access the account. Then, the program checks for sufficient budgeted funds in the account. If there are insufficient funds, the program notifies the submitter, providing the option to continue or to deny the requisitioned item.

When requirements for a submitted requisition have been verified, the submit logic does the following:

- Adds records to the Approval table for the individuals who must approve the requisition
- Notifies the appropriate approvers that they have a requisition to approve

Program Parameters

Introduction

CX contains parameters for executing the *requisition* program.

Parameter Syntax

You can display *requisition* parameters by entering the following command at the shell prompt: **requisition -,**

The following is the correct usage for running *requisition* from the UNIX shell:

```
Usage: requisition [-T requisition] [-o printer name] [-s station number]
where:  requisition type:  type or purpose of requisition
        printer name:  name of printer
        station number:  number of station for document control
```

Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required.

Parameters

The following lists the parameters for running *requisition*.

-T *requisition type* (optional)
Specifies a default requisition form code, where *requisition type* is the form code you specify. The default form code in *requisition* is RP (representing requisition - purchase order type).

Example: requisition -T RP

-o *printer name* (optional)
Specifies the printer name where you want to print copies of the requisitions. The default is the default printer as defined in the macro PRINTER_DEF, or the first printer listed in the CARSPRINTERS list.

Example: requisition -o laser

-s station number (optional)
Specifies the number of the station used to enter and process requisitions. The number is used for document control, and the default is 1.

Example: requisition -s 1

Program Screens

Introduction

The *requisition* program uses six screens to collect and display requisition information.

Access

The screen files for *requisition* are located in the following directory path:
$CARSPATH/modules/requisition/progscr/requisition

Screen Files and Table/Record Usage

The *requisition* screens appear in the following files and use the indicated tables and records:

commlist
Contains the Commodity Code Listing screen
Table/Records: commc_table

reqacct
Contains the Line Item Amount Entry screen
Table/Records: reqacct_rec, subb_table, subt_table

reqhead
Contains the Requisition Header Entry screen
Table/Records: reqhead_rec, id_rec, bldg_table, facil_table, reqaddr_rec, st_table, cntry_table, reqhead_blob

reqline
Contains the Requisition Line Item Entry screen
Table/Records: reqhead_rec, subt_table, reqacct_rec, subb_table, reqline_rec

reqlist
Contains the Listing of Queried Requisitions screen
Table/Records: reqhead_rec

reqparms
Contains the Requisition Parameter Entry screen
Table/Records: form_table

reqsumm
Contains the Requisition Summary screen
Table/Records: reqhead_rec, reqline_rec

SECTION 17 - MENUS, SCREENS, SCRIPTS AND REPORTS

Overview

Introduction

This section provides reference information on the following features of the Requisitioning, Purchasing and Accounts Payable products:

- Menu source files
- Menu option files
- PERFORM screens
- SQL scripts
- Csh scripts
- ACE reports

Directory Locations

The features detailed in this section are located in the following directory paths:

- Menu source files:
  - $CARSPATH/menusrc/fiscal/aprecv
  - $CARSPATH/menusrc/fiscal/requisition
  - $CARSPATH/menusrc/fiscal/requisition/reports
- Menu option files:
  - $CARSPATH/menuopt/purchase
  - $CARSPATH/menuopt/requisition
- PERFORM screens:
  - $CARSPATH/modules/purchase/screens
  - $CARSPATH/modules/requisition/screens
- Scripts:
  - $CARSPATH/modules/ accounting/scripts
  - $CARSPATH/modules/acctspay/scripts
- ACE reports:
  - $CARSPATH/modules/ accounting/reports
  - $CARSPATH/modules/acctspay/reports
  - $CARSPATH/modules/purchase/reports
  - $CARSPATH/modules/requisition/reports

Menu Structure

Introduction

CX menus provide access to a functionally related group of menu options. For example, a CX user in the Admissions office can access all the options necessary for processing prospects, recruits, and applicants from his/her menu, but typically cannot access options for processing accounting information. Depending on the work responsibilities of the CX users at your institution, you can customize their menus to offer access to as many or as few menu options as desired.

The selections that appear on the CX menus at your institution are controlled by a variety of interrelated directories and files. This section explains how the directories and files define CX menus, and it also provides information about the standard CX menu options.

The CX menu source (menusrc) directory path contains definitions of the CX menu structure. Specifically, the $CARSPATH/menusrc/fiscal/aprecv directory path and its subdirectories contain definitions for the Purchasing and Accounts Payable menus.

Directories and Files that Define Menu Structures

The directories and files that control the standard CX menus are:

- Menu source directories
- Menu description files
- Menu option files

Menu Source Directories

Menu source (menusrc) directories define the branches of menus and submenus that offer access to CX features. The highest level of the menu source directory, located at $CARSPATH/menusrc, contains files and subdirectories similar to the following:

```
admit/                        student/
fiscal/                       system/
instdev/                      utility/
menudesc
```

Note: In this example, six subdirectories (designated with a slash [/] after their names) and one file appear in the menusrc directory.

The Menu Description File: Example 1

Menusrc directories always contain a menu description (menudesc) file. The menudesc file defines the options available at the current menu level. To continue the example above, the menudesc file in the $CARSPATH/menusrc directory contains the following type of information:

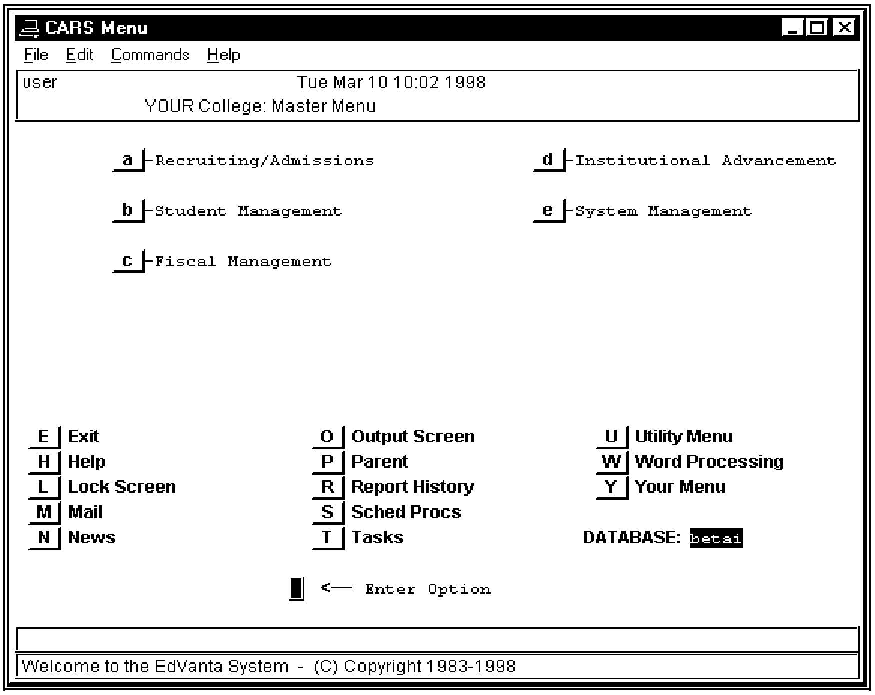
```
#
#Revision Information (Automatically maintained by 'make' - DON'T CHANGE)
#-------------------------------------------------------------------------
#$Header:
#$Log:
#
#-------------------------------------------------------------------------
#
#
TI=INST_NAME: Master Menu
SD=CARS Solution Master Menu
LD=Master Menu For INST_NAME
m4_keepif(ENABLE_MOD_ADMISSIONS, `Y')
MNU_SUB(admit)
m4_keepend
MNU_SUB(student)
MNU_SUB(fiscal)
m4_keepif(ENABLE_MOD_DEVELOP, `Y')
MNU_SUB(instdev)
m4_keepend
MNU_SUB(system)
```

Interpreting the menudesc File in Example 1

In this example, the menudesc file has the following information:

- Comment lines (lines beginning with a pound sign [#])
- Title and descriptive information (lines beginning with TI [title], SD [short description], or LD [long description])
- Keepif lines (lines indicating that a particular submenu is available if a corresponding ENABLE macro is set to Y)
- Keepend lines (lines indicating the end of a macro-controlled option)
- MNU_SUB lines (lines showing the name of the submenu)

Specifically, this example defines a menu called <institution name>: Master Menu. Assuming your institution has purchased the Admissions and Development applications and has enabled the ENABLE_MOD_ADMISSIONS and ENABLE_MOD_DEVELOP macros, your Master Menu would contain the following menu options:



```
user                            Tue Mar 10 10:02 1998
                   YOUR College: Master Menu

          a Recruiting/Admissions          d Institutional Advancement

          b Student Management             e System Management

          c Fiscal Management



 E Exit                 O Output Screen         U Utility Menu
 H Help                 P Parent                W Word Processing
 L Lock Screen          R Report History        Y Your Menu
 M Mail                 S Sched Procs
 N News                 T Tasks               DATABASE: betai

                        <— Enter Option

Welcome to the EdVanta System  -  (C) Copyright 1983-1998
```

Note: The names used for each of the submenus are controlled in their respective submenus.

Subdirectories in the menusrc Directory

Typically, the menusrc directory for a particular menu also contains subdirectories that relate to that menu's submenu options. In the example of the Master Menu, the following subdirectories are in the main $CARSPATH/menusrc directory, and they relate directly to the menu options that appear on the preceding menu screen example.

admit/
Recruiting/Admissions submenu

fiscal/
Fiscal Management submenu

instdev/
Institutional Advancement submenu

student/
Student Management submenu

system/
System Management submenu

utility/
Utility submenu (accessed from the lower part of the CX menu screen)

Contents of menusrc Subdirectories

The menusrc subdirectories have the same components as the main menusrc directory used in the preceding examples. Each contains subdirectories for more submenus, as well as a menudesc file that defines the appearance and contents of the submenu itself.

Navigating the menusrc Subdirectories

By using UNIX commands to move into any of the menusrc subdirectories and viewing its contents (i.e., its own subdirectories, if any, and its related menudesc file), you can view and map the menus, submenus, and menu options available at your institution. To continue the preceding example, if you enter **cd utility** at the UNIX prompt while in the $CARSPATH/menusrc directory, you can then enter **lsf** to display the contents of the utility subdirectory, as in the following:

```
Makefile            data/            iq/              ltrlblrep/       spooler/
RCS/                document/        login/           menudesc
adr/                files/           ltrlbl/          sbscr/
```

If you then enter **cd data** at the UNIX prompt and execute the **lsf** command again, you display the contents of the data subdirectory, as in the following:

```
Makefile            RCS              menudesc
```

In this example, only a menudesc file appears, indicating that no submenus exist at the $CARSPATH/menusrc/utility/data menu source level. The menudesc file in this case contains the actual menu options you can run from the Utilities/Data Entry menu. If desired, you can view the menudesc file using *vi*, *more*, or other UNIX commands.

The Menu Description File: Example 2

The menudesc file in Example 1 contains references only to submenus, that is, each line in the menudesc file that begins with MNU_SUB refers to a submenu. Another type of line can appear in the menudesc file: the MNU_OPT line. MNU_OPT lines do not refer to submenus; instead, they provide the link from the menu to the actual process (e.g., a program, script, report, or screen) you can run. For example, the menudesc file in the $CARSPATH/menusrc/utility/data directory contains the following type of information:

```
#
#Revision Information (Automatically maintained by 'make' - DON'T CHANGE)
#-------------------------------------------------------------------------
#$Header: menudesc,v 8.1 97/10/03 15:34:36 jdoe Released $
#-------------------------------------------------------------------------
#
TI=Utilities: Data Entry Menu
SD=Data Entry
LD=Data Entry
PW=@STD
MNU_OPT(common/programs/ide)
```

Interpreting the menudesc File in Example 2

In this example, the menudesc file has the following information:

- Comment lines (lines beginning with a pound sign [#])
- Title and descriptive information (lines beginning with TI [title], SD [short description], or LD [long description])
- Password information (the line beginning with PW)
- MNU_OPT line (the line showing the location of the menu option itself)

Note:

- As with the menudesc file in Example 1, this file also can contain Keepif and Keepend lines to control the availability of optional processes.
- Although the menudesc file in Example 2 contains only one menu option (common/programs/ide), menudesc files can contain as many options as you care to display on a single menu.

Menu Option Files

The menudesc file for a particular menu or submenu points to the specific menu options you can run (e.g., Example 2 referred to the single menu option common/programs/ide, indicating that a common program is the only option available from the selected submenu).

To locate the menu option files for a particular menu, use the following command syntax:

cd menuopt/<directory on menudesc line>

For Example 2, the appropriate command is:

cd menuopt/common/programs

When you list the contents of the directory, the list resembles the following:

```
Makefile            ctcbatchr        ide              tuserid
RCS/                evnte            ide,fa
adrtest             fo.op            perm.a
ctcbatch            fo.stu           perm.c
```

This directory example contains several menu option files that are linked to other menudesc files. By their names, you usually can determine the process the menu option executes. For example, ide and ide.fa run versions of *identry*, and fo.op and fo.stu run versions of *forment*. More than one version of these processes exist because, depending on the needs of each menu user, you may want to use different titles or parameters.

You can view the specific menu option file using *vi*, *more*, or other UNIX commands.

Interpreting the Menu Option File

The menu option file contains three primary types of information:

- A definition of the parameter screen that appears when a user selects the option from the menu, including information for comment lines
- The name of the screen, program, script, or report the menu option executes
- The parameters, if any, that determine how the menu option should execute

An excerpt from the ide file demonstrates these three types of information. The horizontal lines on the excerpt divide the types of information and have been added to help you interpret the file's contents.

```
-------------------------------- Parameter screen ---------------------------------------
}
screen
{
                    m4_center_clipped(ID DATA ENTRY,40)
                    m4_center_clipped(PP_OFFICE[PA5], 40)
                    m4_center_clipped(PP_TICK[PA7], 40)
}
end
attributes
SD: optional,
    default = "ID Data Entry";
PW: optional,
    default = "@STD";
RD1: optional,
    default = "`Functions:'                          ";
RD2: optional,
    default = "";
RD3: optional,
    default = "`Enter ID Records for Individuals'    ";
RD4: optional,
    default = "Enter ID Records for Non-individuals";
PR: optional,
----------------------------------Executed Process---------------------------------------
    default = "BIN_PATH/identry";
--------------------------------Processing Parameters------------------------------------
m4_keepif(ENABLE_FEAT_AUTOMODE, `Y')
PA1: optional,
    default = "-a";
m4_keepelse
m4_keepif(ENABLE_FEAT_FORCEQUERY, `Y')
PA1: optional,
    default = "-F";
m4_keepend
m4_keepend
PA2: optional,
    default = "-D";
PA3: optional,
    default = "3";
PA4: optional,
    default = "-o";
LU5 = ofc_table.txt, optional;
PA5: optional,
    comments = "COMMENT_OFC_TBCODE  COMMENT_TBL",
    length = 4,
    lookup LU5 joining *ofc_table.ofc,
    upshift;
PA6: optional,
    default = "-T";
LU7 = tick_table.txt, optional;
PA7: optional,
    comments = "COMMENT_TICK  COMMENT_TBL",
    length = 4,
    lookup LU7 joining *tick_table.tick,
    upshift;
end
```

Determining the Type of Executed Process

In the preceding example, the location of the menu option ($CARSPATH/menuopt/common/programs) indicates that the menu option executes a program. The reference to BIN_PATH within the menu option file also indicates the execution of a program. However, many menu options execute other types of processes (e.g., screens, scripts, or reports).

The following list shows the type of process executed, the _PATH reference from the menu option file, and the type of menu option directory path in which the process resides.

Process type	_PATH reference	Menu option directory path
Program	BIN_PATH	/programs
C shell script	SCP_PATH	/scripts
Report	ARC_PATH OTH_PATH	/reports /others
SQL statement	INF_PATH	/informers
PERFORM screen	SCR_PATH FRM_PATH	/screens
Utility	UTL_PATH	/utilities

Summary of Menu Source, Menu Description, and Menu Option Information

When you need to modify any menu options in use at your institution, remember the following:

- Menu source directories reflect your CX menus and submenus.
- Menu source directories contain other menu source subdirectories and menu description files.
- Menu description files define the menu options that appear on each menu.
- Menu description files contain lists of the menu options and submenus.
- Menu option files contain the instructions needed to execute each CX process on every menu.

Requisition and Approval Menus

Introduction

The CX menu structure is defined in the menu source (menusrc) directory path. Specifically, *requisition* and *approval* menus are defined in the $CARSPATH/menusrc/fiscal/requisition directory path. The following directories corresponding to *requisition* and *approval* appear in this path:

- $CARSPATH/menusrc/fiscal/requisition
- $CARSPATH/menusrc/fiscal/requisition/reports
- $CARSPATH/menusrc/system/codes/fin1
- $CARSPATH/menusrc/system/codes/fin1a
- $CARSPATH/menusrc/system/codes/fin2
- $CARSPATH/menusrc/system/codes/fin3
- $CARSPATH/menusrc/system/codes/fin4
- $CARSPATH/menusrc/system/codes/fin5

Each directory above contains a *menudesc* file, which specifies what menu options appear in a menu. Specific menu options, however, are defined in the menu option (menuopt) directory path.

Menu Options

The following list associates each Requisitioning program, screen, and script menu option and corresponding menuopt file and identifies the menuopt locations and what the menu option accesses.

PO/Requisition/Approval

Requisition for Purchase

Accesses: Program: requisition

Menuopt File: $CARSPATH/menuopt/requisition/programs/req.RP

Parameters Passed:

- -t (requisition form type of RP)
- -o (printer)

Requisition from Invoice

Accesses: Program: requisition

Menuopt File: $CARSPATH/menuopt/requisition/programs/req.RI

Parameters Passed:

- -t (requisition form type of RP)
- -o (printer)

Requisition for Check

Accesses: Program: requisition

Menuopt File: $CARSPATH/menuopt/requisition/programs/req.RK

Parameters Passed:

- -t (requisition form type of RP)
- -o (printer)

Req/PO/Inv Approval

Accesses: Program: approval

Menuopt File: $CARSPATH/menuopt/requisition/programs/approval

Parameters Passed: None

Audit Pending Req Amounts

Accesses: Script: updpendreq

Menuopt File: $CARSPATH/menuopt/purchase/scripts/updependreq

Parameters Passed: None

Assign Buyer for Purchase

Accesses: Program: assign

Menuopt File: $CARSPATH/menuopt/purchase/programs/assign

Parameters Passed: None

Purchase Order Entry

Accesses: Program: purchase

Menuopt File: $CARSPATH/menuopt/purchase/programs/purchase

Parameters Passed: None

PO/Invoice Audit Report

Accesses: Program: poinvaudit

Menuopt File: $CARSPATH/menuopt/purchase/programs/poinvaudit

Parameters Passed: None

PO/Invoice Audit - Update

Accesses: Program: poinvaud.u

Menuopt File: $CARSPATH/menuopt/purchase/programs/poinvaud.u

Parameters Passed: None

Vendor Entry

Accesses: Program: vndentry

Menuopt File: $CARSPATH/menuopt/purchasing/programs/vnde

Parameters Passed:

- -f (form selected)
- -a (automatic Query mode)
- -F (force query before insert)
- -D 3 (debug level of 3)

Reports Menu

Requisition Activity

Accesses: ACE Report: requisition/reqactiv

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqactiv

Parameters Passed:
- -f (form type)

Single Requisition

Accesses: ACE Report: requisition/reqapprone

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqapprone

Parameters Passed:
- -f (form type)

Requisitions by Status

Accesses: ACE Report: requisition/reqstat

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqstat

Parameters Passed:
- -f (form type)

Requisition Approvals

Accesses: ACE Report: requisition/reqappr

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqappr

Parameters Passed:
- -f (form type)

Requisition Approvals Outstanding

Accesses: ACE Report: requisition/reqapprout

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqapprout

Parameters Passed:
- -f (form type)

Denied Requisitions

Accesses: ACE Report: requisition/reqdeny

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqdeny

Parameters Passed: None

Requisition to PO Report

Accesses: ACE Report: requisition/reqtpo

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqtpo

Parameters Passed:
- -f (form type)

Purchase Order Approvals

Accesses: Program: poappr

Menuopt File: $CARSPATH/menuopt/purchase/reports/poappr

Parameters Passed:
- -f (form type)

PO Approvals Outstanding

Accesses: Program: poapprout

Menuopt File: $CARSPATH/menuopt/purchase/reports/poapprout

Parameters Passed:
- -f (form type)

Denied POs

Accesses: Program: podeny

Menuopt File: $CARSPATH/menuopt/purchase/reports/podeny

Parameters Passed:
- -f (form type)

PO to Receiving Report

Accesses: ACE Report: requisition/reqtrc

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqtrc

Parameters Passed:
- -f (form type)

Purchase Order Activity

Accesses: ACE Report: requisition/poactiv

Menuopt File: $CARSPATH/menuopt/requisition/reports/poactiv

Parameters Passed:
- -f (form type)

Approval Report for PO

Accesses: ACE Report: requisition/poappr

Menuopt File: $CARSPATH/menuopt/requisition/reports/poappr

Parameters Passed:
- -f (form type)

Purchase Order Aging

Accesses: ACE Report: requisition/poaging

Menuopt File: $CARSPATH/menuopt/requisition/reports/poaging

Parameters Passed:
- -f (form type)

Invoice Aging

Accesses: ACE Report: requisition/invaging

Menuopt File: $CARSPATH/menuopt/requisition/reports/invaging

Parameters Passed:

- -f (form type)

Accounting Table Maintenance: Financial Menus

Approvals by Account

Accesses: Form: taprvact

Menuopt File: $CARSPATH/menuopt/requisition/screens/taprvact

Parameters Passed: None

Approvals by Account Report

Accesses: ACE Report: requisition/taprvact

Menuopt File: $CARSPATH/menuopt/requisition/reports/taprvact

Parameters Passed:

- -f (form type)

Approvals by Comm Code

Accesses: Form: taprvcom

Menuopt File: $CARSPATH/menuopt/requisition/screens/taprvcom

Parameters Passed: None

Approvals by Comm Code Rpt

Accesses: ACE Report: requisition/taprvcom

Menuopt File: $CARSPATH/menuopt/requisition/reports/taprvcom

Parameters Passed:

- -f (form type)

Approvals by ID

Accesses: Form: taprvid

Menuopt File: $CARSPATH/menuopt/requisition/screens/taprvid

Parameters Passed: None

Approvals by ID Report

Accesses: ACE Report: requisition/taprvid

Menuopt File: $CARSPATH/menuopt/requisition/reports/taprvid

Parameters Passed:

- -f (form type)

Commodity Code

Accesses: Form: tcommc

Menuopt File: $CARSPATH/menuopt/requisition/screens/tcommc

Parameters Passed: None

Commodity Code Report

Accesses: ACE Report: requisition/tcommc

Menuopt File: $CARSPATH/menuopt/requisition/reports/tcommc

Parameters Passed:

- -f (form type)

Denial Code

Accesses: Form: tdenial

Menuopt File: $CARSPATH/menuopt/requisition/screens/tdenial

Parameters Passed: None

Denial Code Report

Accesses: ACE Report: requisition/tdenial

Menuopt File: $CARSPATH/menuopt/requisition/reports/tdenial

Parameters Passed:

- -f (form type)

Form Code

Accesses: Form: tform

Menuopt File: $CARSPATH/menuopt/requisition/screens/tform

Parameters Passed: None

Form Code Report

Accesses: ACE Report: requisition/tform

Menuopt File: $CARSPATH/menuopt/requisition/reports/tform

Parameters Passed:

- -f (form type)

Requisition Sort Order

Accesses: Form: sortreq

Menuopt File: $CARSPATH/menuopt/purchase/screens/sortreq

Parameters Passed: None

Req Sort Order Report

Accesses: ACE Report: purchase/sortreq

Menuopt File: $CARSPATH/menuopt/purchase/reports/sortreq

Parameters Passed:

- -f (form type)

Units Code

Accesses: Form: tunits

Menuopt File: $CARSPATH/menuopt/requisition/screens/tunits

Parameters Passed: None

Units Code Report

Accesses: ACE Report: requisition/tunits

Menuopt File: $CARSPATH/menuopt/requisition/reports/tunits

Parameters Passed:

- -f (form type)

Purchasing and Accounts Payable Menus

Introduction

The CX menu structure is defined in the menu source (menusrc) directory path. Specifically, Purchasing and Accounts Payable menus are defined in the $CARSPATH/menusrc/fiscal/aprecv and $CARSPATH/menusrc/fiscal/requisition directory paths. The following directories corresponding to Purchasing and Accounts Payable applications appear in this path:

$CARSPATH/menusrc/fiscal/aprecv
Contains menu options for *receive* (Receiving), *massinv,* and *acctspay* (Accounts Payable).

$CARSPATH/menusrc/fiscal/appurch/reports1
Contains some reports for the applications on the Accounts Payable/Receiving menu.

$CARSPATH/menusrc/fiscal/appurch/reports2
Contains some reports for the applications on the Accounts Payable/Receiving menu.

$CARSPATH/menusrc/fiscal/requisition
Contains menu options for *purchase* (Purchasing) and *assign* (Assign Buyer).

$CARSPATH/menusrc/fiscal/requisition/reports
Contains all the reports for the applications on the PO/Requisitioning/Approval menu, including Purchasing and Assign Buyer.
Note: For more information about the reports on these menus, see *Requisitioning, Purchasing and Accounts Payable ACE Reports* in this manual.

Each directory above contains a *menudesc* file, which specifies what menu options appear in a menu. Specific menu options, however, are defined in the menu option (menuopt) directory path.

Menu Options

The following list associates each Purchasing and Accounts Payable program, screen, and script menu option and corresponding menuopt file, and identifies the menuopt locations and what the menu option accesses.

Accounts Payable/Receiving menu

Invoice Entry

Accesses: $CARSPATH/src/purchase/acctspay (Program)

Menuopt File: $CARSPATH/menuopt/purchase/programs/acctspay

Parameters Passed: None

Receiving Entry

Accesses: $CARSPATH/src/purchase/receive(Program)

Menuopt File: $CARSPATH/menuopt/purchase/programs/receive

Parameters Passed: None

Mass Invoice Entry

Accesses: $CARSPATH/src/purchase/massinv (Program)

Menuopt File: $CARSPATH/menuopt/purchase/programs/massinv

Parameters Passed: None

Vendor Entry

Accesses: $CARSPATH/src/purchasing/vndentry (Program)

Menuopt File: $CARSPATH/menuopt/purchasing/programs/vnde

Parameters Passed:

- -f (Formtype)
- -F (Forced Query mode. Passed if the macro ENABLE_FEAT_FORCEQUERY is set to Y)
- -a (Automatic Query mode. Passed if the macro ENABLE_FEAT_AUTOMODE is set to Y)
- -D (Debugging level)

G/L Journal Reports

Accesses: $CARSPATH/modules/accounting/scripts/jrnlgl (Script)

Menuopt File: $CARSPATH/menuopt/accounting/scripts/jrnlg.APPC

Parameters Passed:

- -f (formtype)

S/L Journal Reports

Accesses: $CARSPATH/modules/accounting/scripts/jrnlsl (Script)

Menuopt File: $CARSPATH/menuopt/accounting/scripts/jrnls.APPC

Parameters Passed:

- -f (formtype)

Accounting Query

Accesses: $CARSPATH/src/accounting/acquery. (Program)

Menuopt File: $CARSPATH/menuopt/accounting/programs/acqu.p

Parameters Passed:
- -p (Printer name)

Budget Review

Accesses: $CARSPATH/src/accounting/bgtreview

Menuopt File: $CARSPATH/menuopt/accounting/programs/bgtr

Parameters Passed:
- -y (Fiscal year to be reviewed)
- -m (Menu name
- -f (Formtype)
- -p (Printer name)

Reports [A-M] Menu

Check Register

Accesses: $CARSPATH/modules/accounting/reports/jrnldocreg (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/jrnldoc.ap

Parameters Passed:
- -f (Formtype)

Credit Memo Activity

Accesses: $CARSPATH/modules/purchase/reports/credmem (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/credmem

Parameters Passed:
- -f (Formtype)

Denied Invoices

Accesses: $CARSPATH/modules/purchase/reports/invdeny (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invdeny

Parameters Passed:
- -f (Formtype)

Discounts Missed

Accesses: $CARSPATH/modules/acctspay/reports (ACE Report)

Menuopt File: $CARSPATH/menuopt/acctspay/reports/discmissed

Parameters Passed:
- -f (Formtype)

Discounts Taken

Accesses: $CARSPATH/modules/acctspay/reports/disctaken (ACE Report)

Menuopt File: $CARSPATH/menuopt/acctspay/reports/disctaken

Parameters Passed:
- -f (Formtype)

Document Register

Accesses: $CARSPATH/modules/accounting/reports/docreg (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/docreg.ap

Parameters Passed:
- -f (Formtype)

Invoice Activity

Accesses: $CARSPATH/modules/purchase/reports/invactiv (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invactiv

Parameters Passed:
- -f (Formtype)

Invoice Aging

Accesses: $CARSPATH/modules/acctspay/reports/invaging (ACE Report)

Menuopt File: $CARSPATH/menuopt/acctspay/reports/invaging

Parameters Passed:
- -f (Formtype)

Invoice Approvals

Accesses: $CARSPATH/modules/purchase/reports/invappr (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invappr

Parameters Passed:
- -f (Formtype)

Invoice Apprvl Outstanding

Accesses: $CARSPATH/modules/purchase/reports/invapprout (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invapprout

Parameters Passed:
- -f (Formtype)

Invoice Forecasting

Accesses: $CARSPATH/modules/acctspay/reports/payfore

Menuopt File: $CARSPATH/menuopt/acctspay/reports/payfore

Parameters Passed:
- -f (Formtype)

Invoices Unsubmitted

Accesses: $CARSPATH/modules/purchase/reports/invunsub (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invunsub

Parameters Passed:
- -f (Formtype)

Invoices With No Checks

Accesses: $CARSPATH/modules/purchase/reports/invnochk (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invochk

Parameters Passed:
- -f (Formtype)

Reports [N-Z] Menu

Open Encumbrances

Accesses: $CARSPATH/modules/purchasing/reports/openpoenc (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchasing/reports/openpoenc

Parameters Passed:
- -f (Formtype)

PO Invoice Receipt Verify

Accesses: $CARSPATH/modules/purchase/reports/invrcpt (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/invrcpt

Parameters Passed:
- -f (Formtype)

PO to Receiving Report

Accesses: $CARSPATH/modules/requisition/reports/potrc (ACE Report)

Menuopt File: $CARSPATH/menuopt/requisition/reports/potrc

Parameters Passed:
- -f (Formtype)

Purchase Order Aging

Accesses: $CARSPATH/modules/acctspay/reports/poaging (ACE Report)

Menuopt File: $CARSPATH/menuopt/acctspay/reports/poaging

Parameters Passed:
- -f (Formtype)

Received Activity

Accesses: $CARSPATH/modules/purchase/reports/recactiv (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/recactiv

Parameters Passed:
- -f (Formtype)

Received Single Item

Accesses: $CARSPATH/modules/purchase/reports/recone(ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/recone

Parameters Passed:
- -f (Formtype)

Requisition to PO Report

Accesses: $CARSPATH/modules/requisition/reports/reqtpo (ACE Report)

Menuopt File: $CARSPATH/menuopt/requisition/reports/reqtpo

Parameters Passed:
- -f (Formtype)

Returned Activity

Accesses: $CARSPATH/modules/requisition/reports/retactiv (ACE Report)

Menuopt File: $CARSPATh/menuopt/requisition/reports/retactiv

Parameters Passed:
- -f (Formtype)

S/L Account Balances

Accesses: $CARSPATH/modules/accounting/reports/subbalance (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/subbal.AP

Parameters Passed:
- -f (Formtype)

S/L Cash Entries

Accesses: $CARSPATH/modules/accounting/reports/subentcash (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/subec.AP

Parameters Passed:
- -f (Formtype)

Subsidiary History

Accesses: $CARSPATH/modules/accounting/reports/subtrhist (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/subtrh.AP

Parameters Passed:
- -f (Formtype)

Vendor Listing

Accesses: $CARSPATH/modules/accounting/reports/vndlst (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/vndlst

Parameters Passed:
- -f (Formtype)

Vendor Spending

Accesses: $CARSPATH/modules/purchase/reports/vndspend (ACE Report)

Menuopt File: $CARSPATH/menuopt/purchase/reports/vndpspend

Parameters Passed:
- -f (Formtype)

1099-R Menu

Initialize Data

Accesses: $CARSPATH/src/acctspay/r1099bld (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/r1099bld

Parameters Passed:
- -y (Academic year)

Edit 1099-R Data

Accesses: $CARSPATH/modules/acctspay/screens/r1099form (PERFORM Screen)

Menuopt File: $CARSPATH/menuopt/acctspay/screens/r1099

Parameters Passed:
- None

Update 1099 Field

Accesses: $CARSPATH/modules/acctspay/screens/upd1099 (PERFORM Screen)

Menuopt File: $CARSPATH/menuopt/acctspay/upd1099

Parameters Passed:
- None

Prepare 1099-R Forms

Accesses: $CARSPATH/src/acctspay/r1099form (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/r1099fm

Parameters Passed:
- -y (Academic Year)

Print Individual Forms

Accesses: $CARSPATH/src/acctspay/r1099 (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/r1099fm.i

Parameters Passed:
- -y (Academic year)
- -l (Default student accounts subsidiary)

Print 1099-R Forms

Accesses: $CARSPATH/src/util/fps (Utility program)

Menuopt File: $CARSPATH/menuopt/utilities/programs/fps.1099r

Parameters Passed:
- None

1099 Menu

Intialize Data

Accesses: $CARSPATH/src/acctspay/f1099bld (Program)

Menuopt File: $CARSPATH/modules/acctspay/programs/99bd

Parameters Passed:
- -y (Academic year)
- -s (Session)

Edit Data

Accesses: $CARSPATH/src/acctspay/f1099form (PERFORM Screen)

Menuopt File: $CARSPATH/modules/acctspay/screens/f1099

Parameters Passed:
- None

Update 1099 Field

Accesses: $CARSPATH/modules/acctspay/upd1099 (PERFORM Screen)

Menuopt File: $CARSPATH/menuopt/acctspay/screens/upd1099

Parameters Passed: None

Prepare Forms

Accesses: $CARSPATH/src/acctspay/f1099 (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/99fm

Parameters Passed:
- None

Prepare Individual Forms

Accesses: $CARSPATH/src/acctspay/f1099form (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/99fm.i

Parameters Passed:
- -l (Default student accounts subisidiary)

Print Forms

Accesses: $CARSPATH/src/util/fps (Utility Program)

Menuopt File: $CARSPATH/menuopt/utilities/programs/fps.1099

Parameters Passed:
- None

Prepare Tape

Accesses: $CARSPATH/src/acctspay/f1099tape (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/99ta

Parameters Passed:
- None

Write Tape

Accesses: $CARSPATH/src/acctspay/f1099tp (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/99tp

Parameters Passed:
- None

Verify Tape

Accesses: $CARSPATH/src/util/tape (Utility Program)

Menuopt File: $CARSPATH/menuopt/utilities/programs/tprd

Parameters Passed:
- None

Prepare Diskette File

Accesses: $CARSPATH/src/acctspay/f1099tape (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/99dsk

Parameters Passed:
- -p *PROG* (Program)
- -d (Display-only mode)

Purchasing/AP Check Writing menu and Refund Check Writing menu

Select Check Group

Accesses: $CARSPATH/src/acctspay/screens/ckgrp (PERFORM screen)

Menuopt File: $CARSPATH/menuopt/accounting/screens/ckgrp

Parameters Passed:
- -f (Formtype)

Select Checks

Accesses: $CARSPATH/src/acctspay/grpselect (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/cksl.d

Parameters Passed:
- -d (Document code of PO)

Select State Grp Sheet

Accesses: $CARSPATH/src/acctspay/ckslct (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/cksl.t

Parameters Passed:
- -d (Display-only mode)
- -t (Required pay terms)

Print Checks

Accesses: $CARSPATH/src/util/fps (Program)

Menuopt File: $CARSPATH/menuopt/utilities/programs/fps.ck

Parameters Passed:
- None

Post Checks

Accesses: $CARSPATH/src/programs/ckpost (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/apckpost

Parameters Passed:
- -g (record status of W)

Select Grouping Sheets

Accesses: $CARSPATH/src/acctspay/grpselect (Program)

Menuopt File: $CARSPATH/menuopt/acctspay/programs/grpsl

Parameters Passed: None

Enter Manual Checks

Accesses: $CARSPATH/src/accounting/voucher (Program)

Menuopt File: $CARSPATH/menuopt/accounting/programs/vch.QC

Parameters Passed:
- -v (journal type of QC)

Check Register

Accesses: $CARSPATH/modules/accounting/jrnldocreg (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/jrnldoc.ap

Parameters Passed:
- -f (formtype)

Grouping Sheet Reconciliation

Accesses: $CARSPATH/modules/accounting/reports/grrecon (ACE Report)

Menuopt File: $CARSPATH/menuopt/accounting/reports/grprecon

Parameters Passed:
- None

G/L Journal Reports

Accesses: $CARSPATH/modules/accounting/scipts/jrnlgl (Script)

Menuopt File: $CARSPATH/menuopt/accounting/scripts/jrnlgl.AP

Parameters Passed:
- -f (formtype)

S/L Journal Reports

Accesses: $CARSPATH/modules/accounting/scripts/jrnlgl (Script)

Menuopt File: $CARSPATH/menuopt/accounting/scripts/jrnlsl.AP

Parameters Passed:

- -f (formtype)

Void Checks

Accesses: $CARSPATH/modules/accounting/programs/docvoid

Menuopt File: $CARSPATH/menuopt/accounting/programs/docv.CK

Parameters Passed:

- None

Purchasing and Accounts Payable PERFORM Screens

Introduction

Purchasing and Accounts Payable uses PERFORM screens for displaying maintenance tables and some records. You can access the screen files in the following directory: $CARSPATH/modules/purchase/screens

PERFORM Screens

The following list contains the PERFORM screens used in Purchasing and Accounts Payable.

- $CARSPATH/menuopt/purchase/screens/sortreq
- $CARSPATH/menuopt/purchase/screens/tbuyer
- $CARSPATH/menuopt/purchase/screens/tcommod
- $CARSPATH/menuopt/purchase/screens/tcond
- $CARSPATH/menuopt/purchase/screens/tretreas
- $CARSPATH/menuopt/purchase/screens/tshpmeth

Screen Reports

A file containing a report for each of the tables appears in the following directory path: $CARSPATH/menuopt/purchase/reports

Purchasing and Accounts Payable Scripts

Introduction

Purchasing and Accounts Payable contains scripts to perform queries, complete processes, and produce reports of database records.

Note: ACE reports and CSH scripts can use SQL scripts in their parameters. Such SQL scripts do not reside on the CX menu system.

Scripts

The following scripts relate to Purchasing and Accounts Payable.

Accounts Payable/Receiving menu: G/L Journal Reports/ S/L Journal Reports

Script file: accounting/scripts/jrnlg.APPC
Description: Prints *jrnlent* and *jrnlacct*, listings of a specified journal with related entries and transactions.
Tables used:

- General Ledger Entry record (gle_rec)
- General Ledger Transaction record (gltr_rec)

Check Writing: Problem Solving menu: Interrupted Posting

Script file: acctspay/scripts/intpost
Description: Corrects the results from the check posting process if the system encountered a problem during a check posting session
Tables used:

- Check Request record (ckreq_rec)
- Journal record (vch_rec)
- Subsidiary Balance record (subb_rec)

Check Writing: Problem Solving menu: Prepare for Check Abort

Script file: acctspay/scripts/remtrk
Description: Removes the tracking file for a specified check group.
Note: Use this option when you must abort a check run after printing begins.
Tables used: None

Check Writing: Problem Solving menu: Prepare for FPS Restart

Script file: acctspay/scripts/restform
Description: Restores form files to an in-progress printing state
Tables used: None

Check Writing: Problem Solving menu: Terminate Check Journal

Script file: acctspay/scripts/termpost
Description: Restores the check writing process to a state in which the checks can be re-posted
Tables used:

- Check Group record (ckgrp_rec)
- Document table (doc_table)
- Subsidiary Balance record (subb_rec)
- Document table (doc_table)

PO/Requisition/Approval menu: Audit Pending Req Amounts

Script file: purchase/scripts/updpendreq

Description: Serves as audit process to ensure pendreq_recs and their totals are correct.

Record used:

- Pending Requisition record (pendreq_rec)

Purchasing/AP: Check Writing menu: G/L Journal Reports
Refund: Check Writing menu: G/L Journal Reports

Script file: accounting/scripts/jrnlgl.AP

Description: Prints *jrnlent* and *jrnlacct*, listings of a specified journal with related entries and transactions.

Tables used:

- General Ledger Transaction record (gltr_rec)
- Journal record (vch_rec)
- Journal table (vch_table)

Requisitioning, Purchasing and Accounts Payable ACE Reports

Introduction

CX provides ACE reports for the reporting of Requisitioning, Purchasing and Accounts Payable database information.

Requisitioning ACE reports

The following ACE reports relate to Requisitioning.

Report title: reqactiv
Menu option: Requisition Activity
Report file location: $CARSPATH/modules/requisition/reports/reqactiv
Description: Prints a report of all requisition activity for the specified date range

Report title: reqappr
Menu option: Requisition Approvals
Report file location: $CARSPATH/modules/requisition/reports/reqappr
Description: Prints a report of all requisition approvals for the specified date range

Report title: reqapprone
Menu option: Single Requisition
Report file location: $CARSPATH/modules/requisition/reports/reqapprone
Description: Prints a report of all items on the specified requisition

Report title: reqapprout
Menu option: Req Approvals Outstanding
Report file location: $CARSPATH/modules/requisition/reports/reqapprout
Description: Prints a report of all requisition approvals outstanding

Report title: reqdeny
Menu option: Denied Requisitions
Report file location: $CARSPATH/modules/requisition/reports/reqdeny
Description: Prints a report of all requisition denials for the specified date range

Report title: reqstat
Menu option: Requisitions by Status
Report file location: $CARSPATH/modules/requisition/reports/reqstat
Description: Prints a report of all requisitions by status for the specified date range

Report title: reqtpo
Menu option: Requisition to PO Report
Report file location: $CARSPATH/modules/requisition/reports/reqtpo
Description: Prints a report of all purchase orders containing items from the specified requisition

Purchasing and Accounts Payable ACE reports

The following ACE reports relate to Purchasing and Accounts Payable.

Report title: credmem
Menu option: Credit Memo Activity
Report file location: $CARSPATH/modules/purchase/reports/credmem
Description: Prints a listing of credit memos written during the specified time period

Report title: discmissed

Menu option: Discounts Missed
Report file location: $CARSPATH/modules/acctspay/reports/discmissed
Description: Prints a list (by vendor) of the discounts that the institution missed during the specified time period

Report title: disctaken

Menu option: Discounts Taken
Report file location: $CARSPATH/modules/acctspay/reports/disctaken
Description: Prints a list (by vendor) of the discounts that the institution took during the specified time period

Report title: docreg

Menu option: Document Register
Report file location: $CARSPATH/modules/accounting/reports/docreg
Description: Prints a document register

Report title: invactiv

Menu option: Invoice Activity
Report file location: $CARSPATH/modules/purchase/reports/invactiv
Description: Prints a list of invoices added during the specified time period, including due date, payee, items purchased and amounts

Report title: invaging

Menu option: Invoice Aging
Report file location: $CARSPATH/modules/acctspay/reports/invaging
Description: Prints a list of outstanding invoices, including the number of days the invoice is past due, based on the date specified when running the report

Report title: invappr

Menu option: Invoice Approvals
Report file location: $CARSPATH/modules/purchase/reports/invappr
Description: Prints a report of all invoice approvals for the specified date range

Report title: invapprout

Menu option: Invoice Apprvl Outstanding
Report file location: $CARSPATH/modules/purchase/reports/invapprout
Description: Prints a report of all invoice approvals outstanding

Report title: invdeny

Menu option: Denied Invoices
Report file location: $CARSPATH/modules/purchase/reports/invdeny
Description: Prints a report of all invoice denials for the specified date range

Report title: invnochk

Menu option: Invoices With No Checks
Report file location: $CARSPATH/modules/purchase/reports/invnochk
Description: Prints a list of invoices for which no checks have been issued

Report title: invrcpt

Menu option: PO Invoice Receipt Verify
Report file location: $CARSPATH/modules/purchase/reports/invrcpt
Description: Prints a list of purchase orders, invoices and receipts added during the specified time period

Report title: invunsub

Menu option: Invoices Unsubmitted
Report file location: $CARSPATH/modules/purchase/reports/invunsub
Description: Prints a list of invoices that have not been submitted, by invoice number

Report title: jrnldocreg

Menu option: Check Register
Report file location: $CARSPATH/modules/purchase/reports/jrnldocreg
Description: Prints a document register by journal

Report title: jrnlent

Menu option: G/L Journal Reports
Report file location: $CARSPATH/modules/accounting/reports/jrnlent
Description: Prints a listing of a specified journal with related entries and transactions

Report title: jrnlslent

Menu option: S/L Journal Reports
Report file location: $CARSPATH/modules/accounting/reports/jrnlslent
Description: Prints the subsidiary transactions by journal

Report title: openpoenc

Menu option: Open Encumbrances
Report file location: $CARSPATH/modules/purchasing/reports/openpoenc
Description: Prints a list of all open purchase orders, sorted by vendor and PO#, including accounts/amounts charged and posted from invoices, and showing remaining encumbrance

Report title: payfore

Menu option: Invoice Forecasting
Report file location: $CARSPATH/modules/acctspay/reports/payfore
Description: Prints a list of payables by vendor that are due during the specified time period

Report title: poactiv

Menu option: Purchase Order Activity
Report file location: $CARSPATH/modules/requisition/reports/poactiv
Description: Prints a report of all purchase order activity for the specified date range

Report title: poaging

Menu option: Purchase Order Aging
Report file location: $CARSPATH/modules/acctspay/reports/poaging
Description: Prints a list of purchase orders by vendor, including number of days past due based on the date specified when running the report

Report title: poappr

Menu option: Approval Report for PO
Report file location: $CARSPATH/modules/purchase/reports/poappr
Description: Prints a report of all approval records against purchase orders in the specified purchase order

Report title: poappr

Menu option: Purchase Order Approvals
Report file location: $CARSPATH/modules/requisition/reports/poappr
Description: Prints a report of all purchase order approvals for the specified date range

Report title: poapprout

Menu option: PO Approvals Outstanding
Report file location: $CARSPATH/modules/purchase/reports/poapprout
Description: Prints a report of all purchase order approvals outstanding

Report title: podeny

Menu option: Denied POs
Report file location: $CARSPATH/modules/purchase/reports/podeny
Description: Prints a report of all purchase order denials for the specified date range

Report title: potrc

Menu option: PO to Receiving Report
Report file location: $CARSPATH/modules/requisition/reports/potrc
Description: Prints a report that traces a purchase from purchase order to receiving

Report title: recactiv

Menu option: Received Activity
Report file location: $CARSPATH/modules/purchase/reports/recactiv
Description: Prints a list of goods received during the specified time period, including date received, purchase order number and vendor

Report title: recone

Menu option: Received Single Item
Report file location: $CARSPATH/modules/purchase/reports/recone
Description: Prints detail information about a single received item, including quantity, units, condition, item code, and place delivered

Report title: reqtpo

Menu option: Requisition to PO Report
Report file location: $CARSPATH/modules/requisition/reports/reqtpo
Description: Prints a report that traces a request from requisition to purchase order

Report title: retactiv

Menu option: Returned Activity
Report file location: $CARSPATH/modules/purchase/reports/retactiv
Description: Prints a list of items returned during the specified time period

Report title: subbalance

Menu option: S/L Account Balances
Report file location: $CARSPATH/modules/accounting/reports/subbalance
Description: Prints a list of the balances in subsidiary accounts (actual, encumbered and total)

Report title: subentcash

Menu option: S/L Cash Entries
Report file location: $CARSPATH/modules/accounting/reports/subentcash
Description: Prints a report of the cash detail and/or the summary for subsidiaries with or without Balances (bals) or Totals (tots) for the specified time period

Report title: subtrhist

Menu option: Subsidiary History
Report file location: $CARSPATH/modules/accounting/reports/subtrhist
Description: Prints a list of transactions by date, especially for use prior to a subsidiary archiving

Report title: vndlst

Menu option: Vendor Listing
Report file location: $CARSPATH/modules/acctspay/reports/vndlst
Description: Prints a list of all the vendors used by the institution

Report title: vndspend

Menu option: Vendor Spending
Report file location: $CARSPATH/modules/purchase/reports/vndspend
Description: Prints a list of vendors and the amounts spent during the current and prior year

SECTION 18 - CUSTOMIZING THE REQUISITIONING PROCESSES

Overview

Introduction

This section provides you procedures for setting and installing the features of the Requisitioning product. The following procedures are included:

- Assessing institution needs for product
- Reviewing data in tables and records

Basic Information

This section contains detailed procedures specific to the Requisitioning product. For information on performing basic procedures, such as using the MAKE processor and reinstalling options, refer to the following resources:

- Database Tools and Utilities course notebook
- *Jenzabar CX Technical Manual*

Implementation Sequence

Institutions must implement the Requisitioning product after the Purchasing and Accounts Payable product. Requisitions, either approved or denied, are meaningful only when they provide input to the Purchase Order and Accounts Payable functions at the institution.

Assessing the Requisitioning Setup

Introduction

CX provides several ways to implement its options. After assessing the needs of your institution, you can change the menu options in the Requisitioning product to control the options, reports and features that the menu users can see and use.

Requisitioning only uses macros for identifying form types. You control access to all the features of the Requisitioning product by modifying the menu options.

Default Setup

The Requisitioning product's default setup provides options in the Fiscal Management: PO/Requisition/Approval Menu for entering and approving requisitions. If the institution does not want CX users at the institution to have such options, you must modify the settings on the menu.

Reviewing the Setup

After assessing features of *requisition* and *approval* and setting the appropriate menu options, you must review the setup of CX tables, records, and features.

Review the following steps during the setup process at your institution.

Note: Because of the close relationship among all the procurement programs (e.g., *requisition*, *approval*, *purchase*, and *acctspay*), these steps affect both the Requisitioning and the Purchasing/Accounts Payable applications. For more information about customizing Purchasing and Accounts Payable, see *Customizing the Purchasing and Accounts Payable Processes* in this manual.

Step 1

Carefully assess the permissions required to request, approve, view, and pay for goods and services. Since *approval*, *purchase*, and *acctspay* provide access to the Budget Review application, consider which employees should be able to view confidential information (e.g., payroll data). Tables that control these permissions include:

- glperm_table
- perm_table
- userid_table

Note: For more information about these tables, see *General Ledger Technical Manual*.

Step 2

- For each Requisitioning table, review the codes supplied with CX.
- Determine whether or not the codes meet the needs of your institution.
- Make updates as appropriate, ensuring that authorizations provide the level of control required by your institution.

Step 3

- For each Requisitioning screen, review the fields supplied with CX.
- Determine whether or not all the fields are necessary to meet the needs of your institution.
- Make updates as appropriate.

Tables to Review or Modify

Introduction

Because CX is table-driven, correct completion of the tables is essential to the implementation process. This section contains descriptions of both the required and optional tables that support the Requisitioning product.

Authorization Tables

Each institution must customize the authorization tables to define the relationships between requisitions, purchase orders and invoices, and their approver(s). CX uses three tables to define the types of approvals: by account, by ID and by commodity code. In each of these tables, the Dollar Amount field defines the minimum amount to be approved and the method of approval (vertical or horizontal). Additionally, in each table you enter a Y or N in the document approval field to define the document to which the approval applies (requisition, purchase order or invoice).

Primary and Alternate Approvers

Depending on your setup of the authorization tables, the Requisitioning, Purchasing and Accounts Payable products can support primary and alternate approvers at the ID, account, and commodity levels. When an individual is an alternate approver, the document is flagged with an asterisk (*) on the Requisition Header Entry screen (Approval). Regardless of whether an approver is primary or alternate, the first approver to access, review, and approve or deny a selected document will control the approval or denial of the document.

Example: Individual A submits a document requiring the approval of Approver 1, with Approver 2 as the alternate approver. On Monday, Approver 2 reviews documents requiring approval, and denies individual A's document. The document will not appear on the Requisition Header Entry screen (Approval) when Approver 1 accesses it on Tuesday.

Note: If both approvers access the Requisition Header Entry Screen (Approval) at the same time and attempt to approve or deny a document simultaneously, the system will accept the first approver's approval or denial, then display a message to the second approver stating that another approver has already approved or denied the document

Account Authorization Table

The Account Authorization table (auth_acct_table) links approvers to specific accounts or a range of accounts. For example, to indicate that the approver whose CX ID is 53124 must approve all expense account (6000 object number series) requisitions, purchase orders and invoices for the Biology department (function 1003) beginning at $100.00, enter the following values in the auth_acct_table.

Note: In this example, the approver whose CX ID is 31224 is the alternate approver for these accounts, and documents of $100.00 or more that are charged to accounts in the specified range will appear on both the primary and the alternate's Requisition Header (Approval) screens.

Note: You must specify exact account numbers in the account fields. The *approval* program does not recognize wild cards of any kind in the account fields.

Beginning Fund:	10
Beginning Function:	1003
Beginning Object:	6000
Beginning Subfund:	
Ending Fund:	10
Ending Function:	1003
Ending Object:	6999
Ending Subfund:	
Authorization ID:	53124
Alternate Auth ID:	31224
Dollar Amount	100.00
Requisition Approval	Y
PO Approval	Y
Invoice Approval	Y

Note: The system tests for the need for account authorization against the totals charged to each account on a requisition, purchase order or invoice. To continue the above example, assume a document contains two line items that both impact account 10-1003-6100. One of the line items is $70, and the other line item is $60. With account authorization, since the two line items appear on the same document and their total exceeds the minimum amount of $100, they require approval. If the two line items appeared on different documents, they would not require approval.

ID Authorization Table

The ID Authorization table (auth_id_table) links approvers to specific IDs. For example, to indicate that the approver whose CX ID is 53124 must approve all requisitions and purchase orders for a user whose CX ID is 68097, enter the following values in the auth_id_table. By setting the Dollar Amount to $0.00, you cause all requisitions and purchase orders to require approval regardless of amount.

Note: In this example, the approver whose CX ID is 31224 is the alternate approver for this ID, and all requisitions and purchase orders submitted by the ID will appear on both the primary and the alternate's Requisition Header (Approval) screens.

• ID Number:	68097
• Authorization ID:	53124
• Alternate Auth ID:	31224
• Dollar Amount	0.00
• Requisition Approval	Y
• PO Approval	Y
• Invoice Approval	N

Note: ID authorization occurs at the document level. For example, if the Dollar Amount above were $100 (instead of 0.00), and the document totaled $120, then authorization would be required.

Commodity Code Authorization Table

The Commodity Code Authorization table (auth_comm_table) links approvers to specific commodity codes. For example, to indicate that the approver whose CX ID is 53124 must approve all requisitions (not purchase orders or invoices) for computers (commodity code 110011-1, any dollar amount), enter the following values in the auth_comm_table.

Note: In this example, the approver whose CX ID is 31224 is the alternate approver for this commodity code, and all requisitions containing this code will appear on both the primary and the alternate's Requisition Header (Approval) screens.

• Commodity Code:	110011-1
• Authorization ID:	53124
• Alternate Auth ID:	31224
• Dollar Amount	0.00
• Requisition Approval	Y
• PO Approval	N
• Invoice Approval	N

Note: Commodity approval is at the line item level. For example, assume the Dollar Amount above is $100 (instead of 0.00), and a document has two items of commodity code 110011-1, one for $95 and one for $200. The $200 line item causes the document to need an approval.

Horizontal approval setup

To set up horizontal approval, use any of the above tables. Create multiple entries with as many authorization IDs as required, using the *same dollar amounts* as the dollar limit.

Vertical approval setup

To set up vertical approval, use any of the above tables. Create multiple entries with as many authorization IDs as required, using *incremented dollar amounts* as the dollar limit.

Buyer Table

The Buyer table (buyer_table) links a buyer to a station number and form type. Only the *assign* program uses the table; unless the institution enables the *assign* program by enabling the macro ENABLE_FEAT_ASSIGN, the Buyer table is not necessary. Values in this table must correspond to entries in the Document table (doc_table) and to the following macros in the macros/custom/financial file:

- REQ_CHK_DOC_REF
- REQ_INV_DOC_REF
- REQ_PO_DOC_REF

Note: For more information about the macros that enable features in Requisitioning, see *Requisitioning Macros and Includes* and *Purchasing and Accounts Payable Macros and Includes* in this manual*.*

Commodity Code Table

The Commodity Code table (commc_table) enables you to enter commodity codes, descriptions and other information about the merchandise that your institution orders. You can also enter an ID number of a suggested vendor. The Commodity Code table is optional if your institution does not use commodity codes when ordering merchandise.

If you use the Commodity Code table, the table must include an entry with a blank code that has an Item Name and Description of “blank description.” This entry enables you to avoid errors when using the Table Lookup feature. Other entries in the table can come from the U.S. Government Commodity Code Listing, from the institution’s personalized list from its own central store, or a combination of the two.

Denial Table

The Denial table (denial_table) contains codes and explanations for denied requisitions. The table is required for institutions using the *approval* program. Users can add or modify Denial table entries for each institution.

Document Table

The Document table defines the document types that the institution uses. It also assigns numbering sequences to the document types, and identifies the journals to be posted to for each document type. The Document table is required.

Institutions customize the Document table during implementation of the General Ledger product.

Note: The Document table values that relate to Requisitioning must correspond to the following macro values in the macros/custom/financial file:

- REQ_CHK_DOC_REF
- REQ_INV_DOC_REF
- REQ_PO_DOC_REF

Form Table

The Form table (form_table) associates forms and form description codes (e.g., RP, RI, RK) with application codes (e.g., RQ). When the institution installs Requisitioning, the Form table contains all the required valid values.

General Ledger Permission Table

The General Ledger Permission table (glperm_table) defines the account number that a user can access. The Permission code that appears in both the Permission table and the General Ledger Permission table links the table entries. Therefore, at the Permission table level, you define access to programs, and at the General Ledger Permission table level, you define access to accounts within the programs.

Institutions customize the General Ledger Permission table during implementation of the General Ledger product.

Units Table

The Units table (units_table) contains the codes and descriptions of the order quantities that the institution uses (e.g., BOXES, EACH). Users can add or modify the Units table if desired. The table is optional.

User Identification Table

The User Identification table (userid_table) links an individual's system-generated ID number with the UNIX user ID number. The Requisitioning product uses UNIX ID numbers to identify requisitioners and approvers, and then uses CX ID numbers to obtain names and addresses for Requisitioning purposes.

To complete the User Identification table, first locate the CX ID number from the ID record for each user. Then, obtain the UNIX ID number by viewing the /etc/passwd file, searching for the login name of the user and noting the first number after the login name.

Example: To view ID information for the UNIX user *jdoe*, search for *jdoe* in /etc/passwd. A line similar to the following exists for every user at the institution:

jdoe:*:101:100:John Doe, Systems Manager, /usr/jdoe:/bin/csh

In this example, the UNIX ID number for *jdoe* is 101.

Note: Enter the user's login name in the User Identification Table to allow the user to receive electronic mail from the Requisitioning, Purchasing, and Accounts Payable modules.

SECTION 19 - CUSTOMIZING THE PURCHASING AND ACCOUNTS PAYABLE PROCESSES

Overview

Introduction

This section provides you procedures for setting and installing the features of the Purchasing and Accounts Payable product. It includes the following procedures:

- Assessing institution needs for the product
- Reviewing data in tables and records
- Changing default field values in macros

Basic Information

This section contains detailed procedures specific to the Purchasing and Accounts Payable product. For information on performing basic procedures such as using the MAKE processor and reinstalling options, refer to the following resources:

- *Database Tools and Utilities course notebook*
- *CX Technical Manual*

Implementation Sequence

Institutions must implement the Purchasing and Accounts Payable product *before* they implement the Requisitioning product. If an institution decides not to use the Requisitioning application, but needs online approvals for purchase order and invoice documents, the Requisitioning product must be implemented to the extent that the approval and denial tables are set up, and the macro to enable the Requisitioning product is set accordingly.

Assessing the Purchasing and Accounts Payable Setup

Introduction

CX provides several ways to implement the options of the Purchasing and Accounts Payable product. After assessing the needs of your institution, you can change the default settings of the Purchasing and Accounts Payable enable macros and reinstall the product.

This section lists and describes the features that you must assess before you can modify the macros.

Enabling Features

CX's default setup enables the following features:

- Assign program
- Purchasing and Accounts Payable product
- Vendor checking
- Simplified invoice processing

The macros that enable the features appear in the macros/custom/financial file.

Enabling State of Illinois Requirements

CX includes some features for account verification that relate directly to reporting requirements for state institutions in Illinois. Other institutions or states may also benefit from their use. Specifically, the features ensure the following:

- The account entered on an invoice (i.e., the account charged with the ACT amount) is within the Major range of the account charged for the encumbrance.
- The same classification code (if any) that relates to the General Ledger account charged for the encumbrance also relates to the actual account used to post the invoice.

Use the following steps to set up these features:

1. Enable both of these related features in the Configuration table.
2. Define the location of the classification code in the account number in the RPA_GL_CLASS_FIELD macro in $CARSPATH/include/custom/rpa.
3. Reinstall src/purchase and src/requisition.

Form Types

The standard Purchasing and Accounts Payable product contains a variety of define macros that set up valid document and form types. If you change the codes for your document or form types in the Document or Form Type table, you must update the corresponding macros in the macros/custom/financial file. For more information about the relationship between form types and macro values, see *Purchasing and Accounts Payable Macros and Includes* in this manual.

Reviewing Data in Tables and Records

After assessing features of Purchasing and Accounts Payable and setting the appropriate macros, you must review the setup of CX tables.

Reviewing the Setup

The following steps enable you to review the values of the CX tables and records.

Step 1

- For each table, review the codes supplied with CX.
- Determine whether or not the codes meet the needs of your institution.
- Make updates as appropriate.

Step 2

- For each Purchasing and Accounts Payable screen, review the fields supplied with CX.
- Determine whether or not all the fields are necessary to meet the needs of your institution.
- Make updates as appropriate.

Tables to Review or Modify

Introduction

The Implementation team must review the following tables to ensure that they include the correct values for Purchasing and Accounts Payable.

Note: Purchasing and Accounts Payable uses other financial tables and records that are either system-maintained or set up during the implementation of General Ledger. For more information about these tables and records, see *Purchasing and Accounts Payable Tables and Records* in this manual, and *General Ledger Technical Manual*.

Buyer Table

The Buyer table (buyer_table) links a buyer to a station number and form type. Only the *assign* program uses the table (i.e., institutions need the Buyer table only if they define the macro ENABLE_FEAT_ASSIGN to “Y”).

To create entries for the Buyer table, institutions can copy the entry for Station 1 to all the other stations, and then enter the names that correspond to each station.

Notes:

- Values in this table must correspond to entries in the Document table (doc_table) and to the following macros in the macros/custom/financial file:
- REQ_CHK_DOC_REF
- REQ_INV_DOC_REF
- REQ_PO_DOC_REF
- REQ_RI_DOC_REF
- REQ_RP_DOC_REF
- If the institution changes any of the definitions for these macros, then you must modify the Buyer table. For example, if the default value for REQ_PO_DOC_REF changes from PO to PX, then the Buyer table entries for the form type (frm) of PO must also change from PO to PX.
- For more information about the interrelationships between the Buyer table and other implementation considerations, see Interrelationships Between Macros and Table Values in this section.

Condition Table

The Condition table (cond_table) defines the valid condition codes for received items. Although the standard CX includes a completed Condition table, users can add to or modify this table as required.

Commodity Category Table

The Commodity Category table (commod_table) defines valid types of expenditures for the Mass Invoice application. If the institution does not use Mass Invoice, the table is not required.

Document Table

The Document table (doc_table) defines the document types that the institution uses. It also assigns numbering sequences to the document types, and identifies the journals to be posted to for each document type. The Document table is required for each of the document types in Purchasing and Accounts Payable.

When institutions implement Purchasing and Accounts Payable, the Document table contains values for stations 0 and 1. The following table lists the form types and stations included in the standard doc_table, and indicates if, during implementation, institutions can add additional station numbers for the form type.

Form	Description	Station numbers	Can more station numbers be added?
RK	Requisition for a check	0,1	Y
RI	Requisition from an invoice	0,1	Y
RP	Requisition for a purchase order	0,1	Y
PP	Standard purchase order	0,1	Y
PO	Open purchase order	0,1	Y
PI	Purchase order from an invoice	0,1	Y
PB	Blanket purchase order	0,1	Y
CP	Change order for a PP form type	0	N
CO	Change order for a PO form type	0	N
CI	Change order for a PI form type	0	N
CB	Change order for a PB form type	0	N
TM	Temporary invoice	1	N
CM	Credit memo	1	N
IV	Invoice	0,1	Y
RV	Receiving form	1	N

Note: For more information about the interrelationships between the Document table and other implementation considerations, see *Interrelationships Between Macros and* Table *Values* in this section.

Fiscal Calendar Record

The Fiscal Calendar record (fscl_cal_rec) defines open posting periods for accounting transactions and subsidiaries. Although the implementation process for General Ledger requires the setup of the fscl_cal_rec, you must create or verify records with the following contents during the implementation of Purchasing and Accounts Payable.

Subsidiary
A/P

Fiscal year
The current year

Amount type
One entry for each of the following types:

- ACT
- ENC

Beginning date
First date of the current fiscal year

Ending date
Last date of the current fiscal year

Opening date
First date on which you want to permit posting for the year

Closing date
Last date on which you want to permit posting for the year

Period number
Set to 0 (zero)

Period
Six position code, defined as follows:

- First four positions are the fiscal year
- Last two positions are 00

Level number
Set to 0 (zero)

Closing balance indicator
Set to A

Other fields
Left blank

Form Table

The Form table (form_table) associates forms and form description codes (e.g., PO, PP and PI) with application codes (e.g., PU). When the institution installs Purchasing and Accounts Payable, the Form table contains all the required valid values. The Form table requires modification under the following conditions:

- If the institution changes a form type (e.g., from PO to PX) in the macro PO_DOC_REF. For this example, the required modification to the Form table is to locate entries for the code PO, and to change the code to PX. The Document table requires similar modification when the form type changes.
- If the institution wants to define the default lead time for the form type. The lead time is the number of days added to the current date to compute the Need By date.

Note: For more information about the interrelationships between the Buyer table and the Forms table, see *Interrelationships Between Macros and Table Values* in this section.

General Ledger Permission Table

The General Ledger Permission table (glperm_table) defines the account number that a user can access. The Permission code that appears in both the Permission table and the General Ledger Permission table links the table entries. Therefore, at the Permission table level, you define access to programs, and at the General Ledger Permission table level, you define access to accounts within the programs.

Institutions customize the General Ledger Permission table during implementation of the General Ledger product.

Requisition Sort Fields Table

The Requisition Sort Fields table (sortreqfld_table) works with the Requisition Sort Order table to define the way you want to display requisitions in the Assign Buyer application, as well as the Purchasing and Accounts Payable applications. To customize these tables, define a code in the Requisition Sort Order table, and then indicate the sort sequence in the Requisition Sort Fields table. The table maintenance screen available from the CX menu lists the fields by which you can sort the requisitions.

Requisition Sort Order Table

The Requisition Sort Order table (sortreq_table) defines sort sequences for requisitions within the Assign Buyer application, as well as the Purchasing and Accounts Payable applications. The institution can have an unlimited number of ways of presenting requisitions on the Assign Buyer screens. To change the default value that appears when users first access the Assign Buyer for Purchase menu option, complete the following steps:

1. Access the following file:

   $CARSPATH/modules/purchase/progscr/assign/abinit
2. Change the default attribute on the sort field to the desired code.
3. Reinstall the *abinit* screen.

Return Reason Table

The Return Reason table (retreason_table) defines valid codes and reasons for the return of merchandise. Institutions can add to or modify this table as desired.

Shipping Method Table

The Shipping Method table (shipmethod_table) defines the valid methods for shipping goods. Institutions can add to or modify this table as desired. The Receiving application accesses this table when users specify the method by which to return defective or unwanted merchandise. If the institution does not use Receiving, the shipmethod_table is not required.

Units Table

The Units table (units_table) contains the codes and descriptions of the order quantities that the institution uses (e.g., BOXES, EACH). Users can add or modify the Units table if desired. The table is optional.

User Identification Table

The User Identification table (userid_table) links an individual's CX -generated ID number with the UNIX user ID number. The Requisitioning product uses UNIX ID numbers to identify requisitioners and approvers, and then uses CX ID numbers to obtain names and addresses for Requisitioning purposes.

To complete the User Identification table, first locate the CX ID number from the ID record for each user. Then, obtain the UNIX ID number by viewing the /etc/passwd file, searching for the login name of the user and noting the first number after the login name.

Example: To view ID information for the UNIX user *jdoe*, search for *jdoe* in /etc/passwd. A line similar to the following exists for every user at the institution:

jdoe:*:101:100:John Doe, Systems Manager, /usr/jdoe:/bin/csh

In this example, the UNIX ID number for *jdoe* is 101.
Note: Enter the user's login name in the User Identification Table to allow the user to receive electronic mail from the Requisitioning, Purchasing, and Accounts Payable modules.

Vendor Quality Table

The Vendor Quality table (venqual_table) defines special qualities about vendors in the Vendor Entry program. If institutions want to track their use of vendors with special qualities, they can add or modify entries in this table.

Vendor Type Table

The Vendor Type table (ventype_table) defines the type of vendor (e.g., an individual with a Social Security number or a business with Tax ID number) in the Vendor Entry program. If institutions want to track this information about the vendors from whom they purchase goods and services, they can add or modify entries in this table.

Interrelationships Between Macros and Table Values

Introduction

The setup of the Buyer table, the Document table, the Forms table and the Purchasing and Accounts Payable macros affects the successful setup of the product. The user must consistently define corresponding values in the tables and macros.

Form Type Setup

The Document table must define a complete set of form types for station “0”; that is, it must contain entries for every purchase order, invoice, requisition, receiving form, change order and credit memo form type. The system uses the Document table entries to determine permissions for posting transactions to the various journals. If any additional station numbers are assigned to either the purchase order form types or the invoice form types, then the corresponding change order form type does not require a station definition in the Document table.

The following table contains an example of the required forms, and the corresponding change order forms if required.

To use this program ...	you must define a form type of ...	and a station number of ...	You must also have an associated change order form type of ...	and a station number of ...
acctspay	IV	0	N/A	N/A
	IV	1	N/A	N/A
	CM	1	N/A	N/A
purchase	PP	0	CP	0
	PP	1		
	PI	0	CI	0
	PI	1		
	PB	0	CB	0
	PB	1		
	PO	0	CO	0
	PO	1		
receive	RV	1	N/A	N/A
requisition	RP	0	N/A	N/A
	RP	1	N/A	N/A
	RI	0	N/A	N/A
	RI	1	N/A	N/A
	RK	0	N/A	N/A
	RK	1	N/A	N/A

Example of Using Default Purchase Order Macros and Table Values

The following situation and corresponding macro and table values illustrate how to set up buyers and form types.

Assumptions:

- Joe Smith is assigned to station 1 and can be assigned requisitions for purchase orders and requisitions from invoices.
- Ann Brown is assigned station 2 and can be assigned requisitions for purchase orders and requisitions from invoices.
- Pat Pecino is assigned station 3 and can be assigned requisitions for purchase orders.
- Ed Nakito is assigned station 4 and can be assigned requisitions for purchase orders.

Note: Ed Nakito can only write purchase orders with the form type of PP, but must still have an entry in the Buyer table (buyer_table) to be assigned the associated RP form type (requisition for purchase order).

Example:
If the REQ_PO_DOC_REF = RP and
PP_DOC_REF = PP and station numbers are 0, 1 and 4 and
PB_DOC_REF = PB and station numbers are 0, 1 and 2 and
PO_DOC_REF = PO and station numbers are 0, 1, 2 and 3,

then the Buyer table needs the following entries:

Form	Station number	Buyer
RP	0	Unassigned
RP	1	Joe Smith
RP	2	Ann Brown
RP	3	Pat Pecino
RP	4	Ed Nakito

If the REQ_INV_DOC_REF = RI and
PI_DOC_REF = PI and station numbers are 0, 1 and 2

then the Buyer table needs the following entries:

Form	Station number	Buyer
RI	0	Unassigned
RI	1	Joe Smith
RI	2	Ann Brown

SECTION 20 - REQUISITIONING MAINTENANCE PROCEDURES

Overview

Introduction

This section provides you with procedures you need in order to maintain the Requisitioning product, including adding new Approval records.

Tasks

This list shows the tasks for maintaining the Requisitioning product.

1. Update the Authorization tables (auth_acct_table, auth_comm_table, auth_id_table) whenever the individuals who approve requisitions change or leave their positions.
2. Update the Chart of Account tables (fund, function, object and subfund) as account numbers at the institution change.

Note: The Chart of Account tables and other accounting records are typically maintained through the General Ledger product. For more information about maintaining the financial portion of CX, see *General Ledger Technical Manual.*

Updating Pending Requisition Amounts

Occasionally, your total pending requisition amounts (i.e., the cost of items that have been requested but not yet approved for purchase) may differ from the totals that appear on your budget screens. If this situation occurs, you can use the Update Pending Requisition Records script (updpendreq) to recompute totals.

You can execute this script from the CX PO/Requisition/Approval menu by selecting Audit Pending Req Amounts, or run it from the UNIX shell in the directory path $CARSPATH/modules/purchase/scripts.

Rolling Requisitions Forward

You may occasionally need to roll requisitions forward to another accounting period. For example, a user may submit a check requisition for the current year (e.g., 9899) when it actually should have been submitted for 9798. The *acctspay* program only shows the requisitions for the fiscal year of the AP journal, based on the post date specified on the parameter screen.

To roll the requisition into the correct year, change the fscl_yr field on the appritem_rec.

SECTION 21 - PURCHASING AND ACCOUNTS PAYABLE MAINTENANCE PROCEDURES

Overview

Introduction

This section provides you with procedures you need in order to maintain the Purchasing and Accounts Payable product.

The Tasks

This list shows the tasks for maintaining the Purchasing and Accounts Payable product.

1. Review and update the tables set up during implementation. Perform the review annually to ensure that the tables contain only useful values.
2. Update the Chart of Account tables (fund, function, object and subfund) as account numbers at the institution change.

Note: The Chart of Account tables and other accounting records are typically maintained through the General Ledger product. For more information about maintaining the financial portion of CX, see *General Ledger Technical Manual.*

SECTION 22 - PROGRAM ERRORS AND CRASH RECOVERY

Overview

Introduction

This section provides you with the following:

- A list of serious and fatal errors
- Crash recovery procedures

Note: Refer to *Using Requisitioning* and *Using Purchasing and Accounts Payable* for a list of the more common status, field error, and warning messages that can appear when menu users execute the programs in the Requisitioning, Purchasing and Accounts Payable products.

Fatal Errors

Fatal errors can occur in the following programs in Purchasing and Accounts Payable:

- Acctspay
- Approval
- Assign
- Invoice (Mass Invoice)
- Purchase
- Receive
- Requisition

Serious and Fatal Errors

Messages - Requisitioning

The following lists alphabetically the serious and fatal error messages that can appear when you are using Requisitioning.

Could not find uid = xxx in userid_table

No record for the user exists in the userid_table. Add a record in the table for each user who needs to access the product.

Select error near line xxx in file hnd.c, SQL error code 100. No error occurred.

The user attempted to user table lookup of a field after entering an incorrect code (e.g., the account number components for a requisition). To correct this error, enter a blank value in the table that the system uses to complete (or validate) the field (e.g., a fund or function code), along with a description of “blank description” (or some other general explanation).

Messages – Purchasing and Accounts Payable

The following lists alphabetically the serious and fatal error messages that can appear when you are using Purchasing and Accounts Payable.

Could not find uid = xxx in userid_table
No record for the user exists in the userid_table. Add a record in the table for each user who needs to access the product.

Troubleshooting

If the requisition process should fail, you can restart the process from the initial input of the requisition. This approach is the simplest way to correct the failure.

Crash Recovery Procedure

Introduction

The recovery from a core dump can require that you complete a succession of reinstalls.

Core Dump Recovery

The following steps enable you to recover from a core dump of a program.

1. Access the program screens directory for the program.

   Example: % cd $CARSPATH/modules/purchase/progscr

   Example: % cd $CARSPATH/modules/requisition/progscr

2. Reinstall each program screen file.

   Example: % make reinstall F=<filename>
   Note: You can also reinstall all of the screens by entering **% make reinstall F=all**.

3. Did the reinstall of the program screens fix the error?
   - If yes, you are done.
   - If no, go to step 4.

4. Access the source code directory of the program.

   Example: % cd src/purchase/acctspay or

   % cd src/requisition/approval or

   % cd src/requisition/requisition

5. Reinstall the source code for the program.

   Example: % make reinstall

6. Did the reinstall of the program source code fix the error?
   - If yes, you are done.
   - If no, go to step 7.

7. In the source code for the program, delete th e old compiled code for the program.

   Example:

   ```
   % make cleanup
   ```

8. Reinstall the program source code.

   Example:

   ```
   % make reinstall
   ```

9. Did the deletion of the old code and reinstallation of the program source code fix the error?
   - If yes, you are done.
   - If no, go to step 10.

10. Review the libraries for the program. In the source code for the program, review the file, *Makefile*. In the file, search for the parameter, ADDLIBS, which identifies the libraries that you must reinstall.

    Example:

    ```
    % vi Makefile

    /ADDLIBS
    ```

11. Reinstall the libraries for the program and reinstall the source for the program.

    Example:

    ```
    % cd <to appropriate library>

    % make reinstall

    % cd src/purchase/acctspay

    % make reinstall
    ```

    Note: You must reinstall the source program to include any library changes.

12. Did the reinstallation of the libraries for the program fix the error?
    - If yes, you are done.
    - If no, call Jenzabar Support Services.

    Note: When a program crashes, it usually sends a mail message to the user who attempted to run the program. If a call to Jenzabar Support Services is required, the user should describe the contents of any mail messages to the Jenzabar Support Services representative. If the program does not send a mail message to the user, then the program crash was probably the result of a system problem.

INDEX